INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

A Fully Convolutional Neural Network for Speech Enhancement

Se Rim Park[1], *Jin Won Lee*[2]

[1]Carnegie Mellon University, USA
[2]Qualcomm Research, USA

serimp@andrew.cmu.edu, jinwonl@qti.qualcomm.com

Abstract

The presence of babble noise degrades hearing intelligibility of human speech greatly. However, removing the babble without creating artifacts in human speech is a challenging task in a low SNR environment. Here, we sought to solve the problem by finding a 'mapping' between noisy speech spectra and clean speech spectra via supervised learning. Specifically, we propose using fully Convolutional Neural Networks, which consist of lesser number of parameters than fully connected networks. The proposed network, Redundant Convolutional Encoder Decoder (R-CED), demonstrates that a convolutional network can be 12 times smaller than a recurrent network and yet achieves better performance, which shows its applicability for an embedded system.

1. Introduction

Denoising speech signals has been a long standing problem. Decades of works showed feasible solutions which estimated the noise model and used it to recover noise-deducted speech [1, 2, 3, 4, 5]. Nonetheless, estimating the model for a babble noise, which is encountered when a crowd of people are talking, is still a challenging task.

The presence of babble noise, however, degrades hearing intelligibility of human speech greatly. When babble noise dominates over speech, aforementioned methods often times will fail to find the correct noise model [6]. If so, the noise-deduction will render distortion in speech, which creates discomforts to the users of hearing aids [7].

Here, instead of explicitly modeling the babble noise, we focus on learning a 'mapping' between noisy speech spectra and clean speech spectra, inspired by recent works on speech enhancement using neural networks [8, 9, 10, 11]. However, the model size of Neural Networks easily exceeds several hundreds of megabytes, limiting its applicability for an embedded system.

On the other hand, Convolutional Neural Networks (CNN) typically consist of lesser number of parameters than FNNs and RNNs due to its weight sharing property. CNNs already proved its efficacy on extracting features in speech recognition [12, 13] or on eliminating noises in images [14, 15]. But upon our knowledge, CNNs have not been tested in speech enhancement.

In this paper, we attempted to find a 'memory efficient' denoising algorithm for babble noise that creates minimal artifacts and that can be implemented in an embedded device: the hearing aid. Through experiments, we demonstrated that CNN can perform better than Feedforward Neural Networks (FNN) or Recurrent Neural Networks (RNN) with much smaller network size. A new network architecture, Redundant Convolutional Encoder Decoder (R-CED), is proposed, which extracts redundant representations of a noisy spectrum at the encoder and map it back to clean a spectrum at the decoder. This can be viewed as mapping the spectrum to higher dimensions, and projecting the features back to lower dimensions.

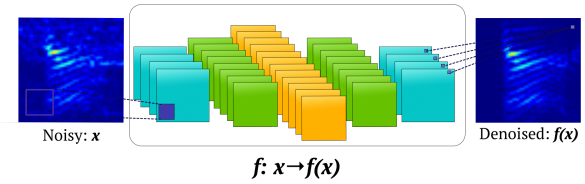

Figure 1: *Speech Enhancement Using a CNN*

The paper is organized as follows. In section 2, a formal definition of the problem is stated. In section 3, the network architectures of proposed R-CED network and comparative networks are presented. In section 4, the experimental methods are specified, and in section 5, the experimental results are presented. In section 6, results are discussed in depth, and in section 7, the study is concluded.

2. Problem Statement

Given noisy spectra $\{\mathbf{x}_t\}_{t=1}^T$ and clean spectra $\{\mathbf{y}_t\}_{t=1}^T$, our aim is to learn a mapping f which generates a segment of 'denoised' spectra $\{f(\mathbf{x}_t)\}_{t=1}^T$ that approximate the clean spectra in ℓ_2 norm, e.g. $\min \sum_{t=1}^T ||\mathbf{y}_t - f(\mathbf{x}_t)||_2^2$. Specifically, we formulate f using a Neural Network (see **Fig.**1). If f is a recurrent type network, the temporal behavior of input spectra is inherently addressed by the network. On the other hand, for a convolutional type network, the past n_T noisy spectra $\{\mathbf{x}_i\}_{i=t-n_T+1}^t$ are considered to denoise the current spectrum, e.g.

$$\sum_{t=1}^{T} ||\mathbf{y}_t - f(\mathbf{x}_{t-n_T+1,\cdots,}\mathbf{x}_t)||_2^2. \quad (1)$$

We set $n_T = 8$, i.e. 8 STFT sequences for input which corresponds to 88ms of speech segment and 1 STFT for output which corresponds to $32ms$ of speech segment (see **Fig.**2 and refer to Sec. 4 for details on STFT parameter settings).

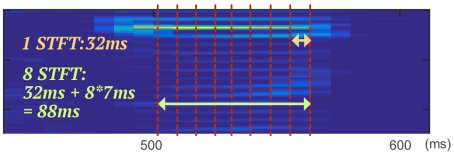

Figure 2: *Input Frames vs. Output Frames*

http://dx.doi.org/10.21437/Interspeech.2017-1465

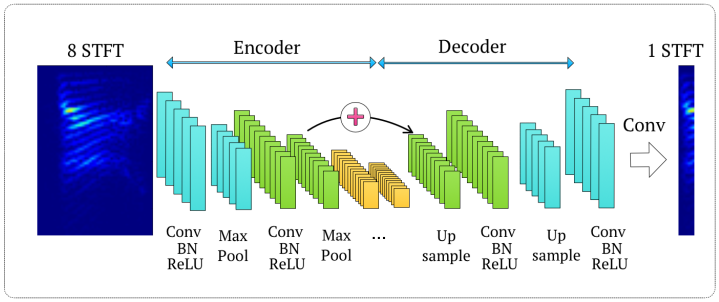

Figure 3: *CED Network*

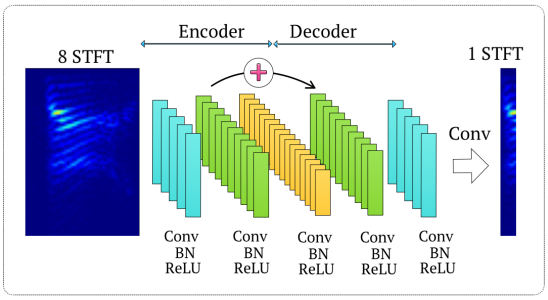

Figure 4: *Proposed Redundant CED (R-CED)*

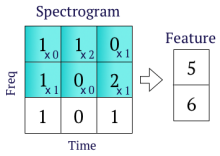

Figure 5: *1-d Convolution*

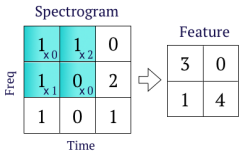

Figure 6: *2-d Convolution*

3. Redundant Convolutional Network

Here, we propose a convolutional network architecture, namely a Redundant Convolutional Encoder-Decoder (R-CED) network, carefully devised for solving a spectrogram regression problem. A R-CED consists of sets of convolution (Conv), batch-normalization (BN) [16], and ReLU [17] layers (see **Fig.**4, each 'block' represents a feature). *No pooling layer is present, and hence no up-sampling layer either.* As opposed to a CED (presented below), R-CED's encoder maps features to a *higher dimension* and the decoder *compresses* features back to the STFT's dimension. The number of filters of are kept symmetric – the number of filters gradually increase at an encoder and decrease at a decoder. The last layer is also convolutional, which makes the R-CED a fully convolutional network.

Convolutional Encoder-Decoder Network (CED) The Conventional Convolutional Encoder-Decoder (CED) network proposed in [18] consists of symmetric encoding layers and decoding layers (see **Fig.**3). An encoder consists of sets of Conv, BN, max-pooling, and ReLU layers. Decoder consists of sets of Conv, BN, and up-sampling layers. A CED *compresses* features along an encoder, and then *reconstructs* features along a decoder. In this paper (and experiments), the final layer of CED is also tuned to be convolutional (as R-CED), which render the CED a fully convolutional network.

Cascaded R-CED Network (CR-CED): The Cascaded Redundant Convolutional Encoder-Decoder Network (CR-CED) is a variation of the R-CED network. It consists of replicates of R-CED Networks stacked vertically. Compared to the R-CED of the same number of parameters, the CR-CED achieves better performance with less convergence time.

Skip Connections: Skip connections [14, 15] are assumed to facilitate optimization in training phase and improve overall performance. Variations of CED, R-CED, and CR-CED with skip connections were compared with original CED, R-CED and CR-CED. Skip connections in [14], which are more suitable for symmetric encoder-decoder design, were utilized. Skip connections are added every other layer. In **Fig.**3 **Fig.**4, skip connections are illustrated and as an 'addition' symbol with an arrow.

1-dim Convolution along freq-axis Every convolution layer in this paper utilizes a 1-dim convolution, i.e. applying convolution only along freq-axis (see **Fig.**5). This turns out to be more efficient than 2-dim convolution for input spectra of size $129(\text{freq}) \times 8(\text{time})$.

4. Experimental Methods

Dataset: The experiment was conducted using the TIMIT database [19] and 27 noise clips from [20]. The noise are mostly multi-talk babble, but also includes variety of noise such as instrumental sounds. Both data in the training set (4620 utterances) and the testing set (200 utterances) were added with one of 27 noise clips at 0dB SNR. 20% of the training features were assigned for the validation.

Feature Transformation: The audio signals were down-sampled to 8kHz, and the silent frames were removed from the signal. Spectral vectors were computed using a 256-point Short Time Fourier Transform (32ms hamming window) with a window shift of 64-point (8ms). 256-point STFT magnitude vectors were reduced to 129-point by removing the symmetric half. 8 STFT magnitude vectors ($\sim 88ms$) were used for input features, and were standardized to have zero mean and a unit variance.

Phase Aware Scaling: At reconstruction, noisy spectral phase was used instead for inverse STFT. To avoid extreme differences (more than 45 degree) between the noisy and clean phase, the clean spectral magnitude was encoded as similar to [21], i.e. $\mathbf{s}_{\text{phase aware}} = \mathbf{s}_{\text{clean}} \cos(\theta_{\text{clean}} - \theta_{\text{noisy}})$. 'Phase aware' STFT magnitude vectors were used for output features, and were standardized to have zero mean and and unit variance.

Training Schemes FNN/CNN filter weights were initialized as [22] and RNN weights were initialized as [23]. FNN and RNN were *pre-trained* with lesser-depth networks first; deeper networks were trained with the weights initialized with the lesser-depth networks' weights. CNNs were trained from scratch. The Adam [24] gradient descent optimization was used for back-propagation with a mini-batch size of 64. The learning rate started from $lr = 0.0015$ with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1.0e^{-8}$. When the validation loss stopped decreasing for more than 4 epochs, learning rate was decreased to $lr/2, lr/3, lr/4$, subsequently. For FNN and RNN, an additional training was performed with ℓ_2 regularization ($\lambda = 10^{-5}$).

Evaluation Score: Signal to Distortion Ration (SDR) [25] was used to measure the amount of ℓ_2 error present between clean and denoised speech: $SDR := 10 \log_{10} ||\mathbf{y}||^2 / ||f(\mathbf{x}) - \mathbf{y}||^2$. In addition, Short time Objective Intelligibility (STOI) [26] and Perceptual Evaluation of Speech Distortion (PESQ)

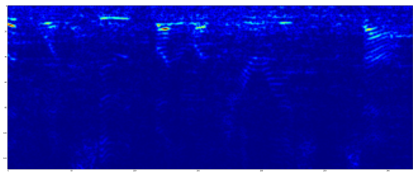

Figure 7: *Noisy Spectrogram*

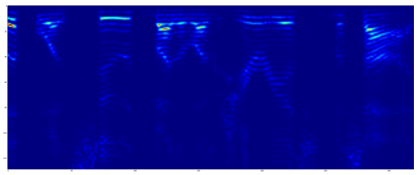

Figure 8: *Clean Spectrogram*

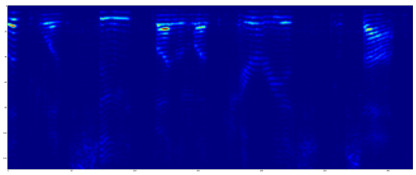

Figure 9: *Denoised Spectrogram*

Table 1: *Network Configurations for CNNs: CED vs. R-CED vs. CR-CED*

	Layer Configuration	Number of Filters	Filter Width
CED, CED /w skip (11 Conv)	Encoder: (Conv, BN, ReLU, Pool) × 5, Decoder: (Conv, BN, ReLU, Upsample) × 5, Conv.	12-16-20-24-32-24-20-16-12-8-1	13-11-9-7-5-7-9-11-13-8-129
R-CED, R-CED /w skip (10 Conv)	(Conv, ReLU, BN) × 9, Conv.	12-16-20-24-32-24-20-16-12-1	13-11-9-7-7-7-9-11-13-129
R-CED, R-CED /w skip (16 Conv)	(Conv, ReLU, BN) × 15, Conv.	10-12-14-15-19-21-23-25-23-21-19-15-14-12-10-1	11-7-5-5-5-5-7-11-7-5-5-5-5-7-11-129
CR-CED (16 Conv)	(Conv, ReLU, BN) × 15, Conv.	(18-30-8) × 5, 1	(9-5-9) × 5, 129

[27] were used to measure the subjective quality of listening.

5. Experimental Results

5.1. Test 1: FNN vs. RNN vs. CNN

In the first experiment, a CNN was compared to a FNN/RNN to demonstrate that convolutional type networks could perform as good as (or even better than) the fully connected type networks. **Table.**2 shows the network configurations that yielded the best performance for each network type. The best performing CNN is 4 times deeper (16 Conv layers) than the best performing FNN/RNN (4 FC/RC layers). **Fig.**10 illustrates the denoising performance of FNN, RNN and CNN (left), and the corresponding number of parameters (right). All network types exhibited similar performance.

Nevertheless, the number of parameters for CNN was about 68 times smaller than that of FNN and 12 times smaller than that of RNN. In addition, the CNN was trained from scratch only once, but FNN/RNN were trained 4 times, subsequently from shallow networks to deeper networks, initialized by the network's weights trained previously. FNN/RNN were also trained once more with ℓ_2 regularization.

	# of Layers	# of Nodes
FNN	(FC, ReLU) × 3, FC	1024-1024-1024
RNN	(RC, ReLU) × 3, FC	256-256-256
CNN (R-CED)	(Conv, ReLU, BN) × 5, Conv	10-12-14-15-19-21-23-25 23-21-19-15-14-12-10-1

Table 2: *Network Config. for FNN vs. RNN vs. CNN*

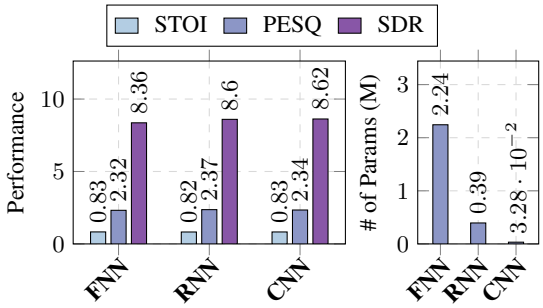

Figure 10: *A Comparison of Denoising Performance and the Network Size for FNN vs. RNN vs. CNN*

In short, this experiment confirms that a CNN can be configured with far lesser number of parameters. Also, because a CNN is much simpler compared to a RNN/FNN, a CNN trains from scratch without any pre-training, whereas an RNN/FNN requires one. Yet, a CNN can achieve similar or better performance.

5.2. Test 2: CED vs. R-CED

In the second experiment, a R-CED was compared to a CED. For a fair comparison, the total number of parameters were fixed to 33,000 while the depth of the networks was fixed to 10 Conv layers. The filter width was configured considering the symmetric encoder-decoder structure (the number of parameters are gradually increased and decreased). Also, the 'frequency coverage' was maneuvered to be equal for both networks [1], i.e. both networks utilized the same number of frequency bins to reconstruct a single frequency bin. Test 2 network config. is summarized at top two rows of **Table.**1.

The denoising performance of CED and R-CED are shown in **Fig.**11 (first 4 bars). Regardless of the presence of skip connections, R-CED yielded better results. The R-CED with skip connections showed the best performance, and the CED without skip connections showed the worst. The effect of the skip connection was prominent in CED (5.96 to 7.92), whereas not so much in R-CED. This effect would be discussed in more detail at Sec. 6.3

5.3. Test 3: Finding the R-CED with the Best Performance

In the third experiment, a variety of R-CED and CR-CED network configurations were tested and compared to find the best performance network. The number of parameters considered were 33K and 100Km and the depth of networks considered were 10, 16, and 20 Conv layers. **Table.**1 (Row 2-4) summarizes the specific network configurations. The denoising performances are presented in **Fig.**11 (five bars from right). A few notable observations are i) the number of network parameters was the most dominant factor that was associated with the network performance, ii) the network depth was secondary, iii) CR-CED with skip connection yielded the best performance

[1] Here a 'frequency coverage' refers to how much nearby frequency bins at the input are used to reconstruct a single frequency bin at the output.

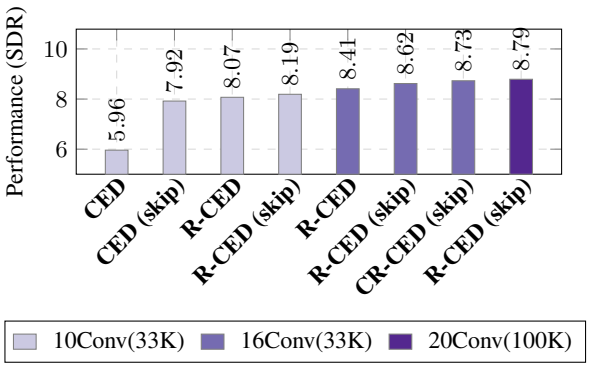

Figure 11: *A Comparison of Denoising Performance and the Model Size for different CNN Architectures*

when other conditions were kept same (16 convolution layers, 33K parameters).

6. Discussion

Here we discuss why the proposed RCED network is effective.

6.1. The Effect of the Number of STFT Input Vectors

We partly accredit the success of denoising (not only for CNN but for FNN/RNN as well) to input-output pairing schemes. The input consists of 8 STFT vectors $\sim 88ms$, and the output consists of 1 STFT vector $\sim 32ms$ (see **Fig.**2). Considering that the average normal length of vowels in speech is around $99ms$ [28], $88ms$ seems to be optimal length to contain informative characteristics of speech. When input vectors of length 4 or 16 were tested, the performance degraded (See **Fig.**12 (left) for the denoising score vs. the size of input STFT vectors).

6.2. Lesser Number of Parameters to Train

Fig.10 (right) shows the number of parameters for each network. The number of parameters of CNN is 68 times smaller than that of the FNN, and 12 times smaller than that of the RNN. Lesser complexity of the network implies that the CNN will be trained with a smaller training set. When the same experiment in Sec.5.1 was repeated but with the half the size of the training set, performance of FNN and RNN began to diminish, whereas the performance of CNN remained solid (see **Fig.**12, right). In other words, FNN/RNN is much prone to overfitting because of higher complexity. Also, because FNN/RNN consists of excessive parameters, they didn't converge when trained from scratch. Therefore, 4-layer FNN/RNN had to be pretrained twice with lesser-depth networks first. Also, FNN/RNN were trained once more involving ℓ_2 regularizations. Therefore, roughly FNN/RNN required 4-times more training compared to the CNN.

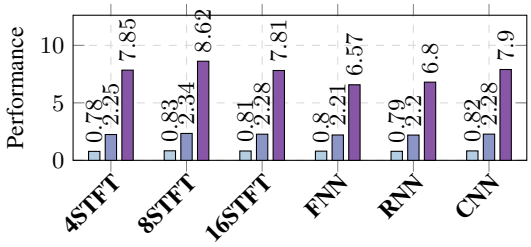

Figure 12: *Performance vs. # of input vectors (left), performance with lesser number of parameters (right)*

6.3. Feature Mapping to Higher Dimension with no Pooling Layers

In Sec. 5.2, we noticed that the performance of R-CED was robust regardless of the skip connection, whereas the performance of CED was affected hugely by the presence of the skip connection. This matches with our intuition that CED's encoder *compresses* features in the max-pooling layers. This results in a 'loss' of information, and thus CED's decoder cannot reconstruct the 'lost' information. However, when skip connections are present, the 'lost' information would be provided from the encoder to the decoder, and thus CED can reliably reconstruct the spectrogram. Without skip connections, resulting speech from CED sounds artificial and mechanical, which implies that the necessary components for audios to sound like human speech are lost at pooling layers, and cannot be reconstructed at the decoder unless additional information is provided (i.e. by skip connections).

On the other hand, the effect of the skip connection was not very notable in R-CED (8.07 to 8.19). This is because the features along R-CED's encoder are augmented with 'redundant' information. This can be viewed as a mapping to a higher dimension. By generating redundant representations of features at the encoder, and removing unwanted features at the decoder, the speech quality could be effectively enhanced.

7. Conclusion

In this paper, we proposed a memory efficient babble denoising method utilizing CNN. Inspired by the past success of FNN and RNN, we hypothesized that CNN can effectively denoise speech with smaller network size according to its weight sharing property. Through experiments, we demonstrated that CNN can yield similar or better performance with much lesser number of model parameters. With smaller network complexity, CNN could be trained faster with lesser data.

Also, we proposed a new fully convolutional network architecture named R-CED, and showed its efficacy in spectrogram regression. We observed that the success of R-CED is associated with the increasing dimension of the feature space along the encoder and decreasing dimension along the decoder. We expect that R-CED can be also applicable to regression problems in other domains. We plan to expand the study further to generalizing the method to different noise levels and unseen noise types, as well as pruning the networks to minimize the convolution operation counts.

8. 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] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[4] P. Scalart *et al.*, "Speech enhancement based on a priori signal to noise estimation," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 629–632.

[5] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on speech and audio processing*, vol. 3, no. 4, pp. 251–266, 1995.

[6] N. Krishnamurthy and J. H. Hansen, "Babble noise: modeling, analysis, and applications," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 7, pp. 1394–1407, 2009.

[7] A. McCormack and H. Fortnum, "Why do people fitted with hearing aids not wear them?" *International Journal of Audiology*, vol. 52, no. 5, pp. 360–368, 2013.

[8] K. Han, Y. Wang, and D. Wang, "Learning spectral mapping for speech dereverberation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4628–4632.

[9] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[10] B. Xia and C. Bao, "Speech enhancement with weighted denoising auto-encoder." in *INTERSPEECH*, 2013, pp. 3444–3448.

[11] K. Osako, R. Singh, and B. Raj, "Complex recurrent neural networks for denoising speech signals," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.

[12] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[13] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[14] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections," *arXiv preprint arXiv:1603.09056*, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[19] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[20] V. Akkermans, F. Font, J. Funollet, B. De Jong, G. Roma, S. Togias, and X. Serra, "Freesound 2: An improved platform for sharing audio clips," in *Klapuri A, Leider C, editors. ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference; 2011 October 24-28; Miami, Florida (USA). Miami: University of Miami; 2011.* International Society for Music Information Retrieval (ISMIR), 2011.

[21] P. Mowlaee and R. Saeidi, "Iterative closed-loop phase-aware single-channel speech enhancement," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1235–1239, 2013.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[23] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv preprint arXiv:1504.00941*, 2015.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[25] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[26] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech." in *ICASSP*, 2010, pp. 4214–4217.

[27] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 749–752.

[28] H. Kuwabara, "Acoustic properties of phonemes in continuous speech for different speaking rate," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4. IEEE, 1996, pp. 2435–2438.