arXiv:1609.07132v1 [cs.LG] 22 Sep 2016

# A FULLY CONVOLUTIONAL NEURAL NETWORK FOR SPEECH ENHANCEMENT

*Se Rim Park*

Carnegie Mellon Univeristy
Pittsburgh, PA, 15213

*Jin Won Lee*

Qualcomm Research
San Diego, CA, 92121

## ABSTRACT

In hearing aids, 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: the hearing aids.

***Index Terms—*** Speech Enhancement, Speech Denoising, Babble Noise, Fully Convolutional Neural Network, Convolutional Encoder-Decoder Network, Redundant Convolutional Encoder-Decoder Network

## 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 (e.g. kernel method), and projecting the features back to lower dimensions.

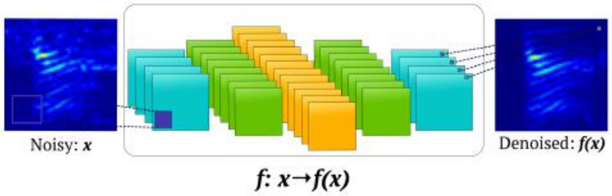

**Fig. 1**: Speech Enhancement Using a CNN

The paper is organized as follows. In section 2, a formal definition of the problem is stated. In section 3, the fully convolutional network architectures are presented including the proposed R-CED network. In section 4, the experimental methods are provided. In section 5, description of the experiments and the corresponding network configurations are provided. In section 6, the results are discussed, and in section 7, we end with conclusion of this study.

## 2. PROBLEM STATEMENT

Given a segment of 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 the $\ell_2$ norm, e.g.

$$\min \sum_{t=1}^{T} ||\mathbf{y}_t - f(\mathbf{x}_t)||_2^2. \tag{1}$$

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 already addressed by the network, and hence objective (1) suffices. 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 spectra, e.g.

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

We set $n_T = 8$. Hence, input spectra to the network is equivalent to about 100ms of speech segment, whereas the output spectra of the network is of duration 32ms (see **Fig.**2, **Fig.**3).

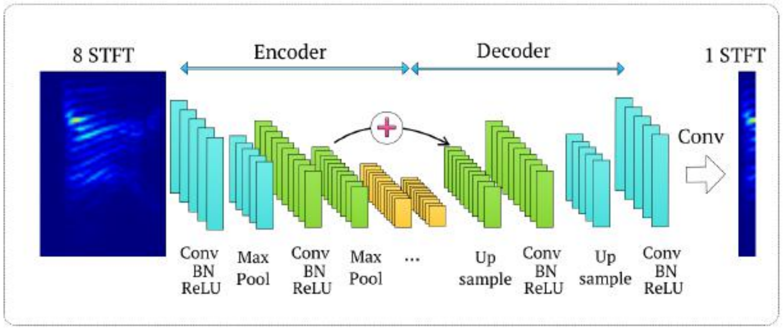

**Fig. 2**: Modified Convolutional Encoder-Decoder Network (CED)

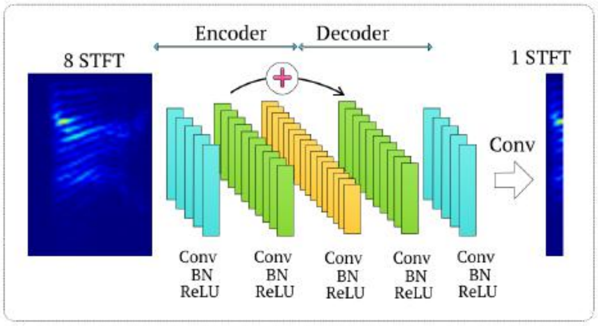

**Fig. 3**: Proposed Redundant CED (R-CED)

## 3. CONVOLUTIONAL NETWORK ARCHITECTURES

### 3.1. Convolutional Encoder-Decoder Network (CED)

Convolutional Encoder-Decoder (CED) network proposed in [16] consists of symmetric encoding layers and decoding layers (see **Fig.**2, each block represents a feature). Encoder consists of repetitions of a convolution, batch-normalization [17], max-pooling, and an ReLU [18] activation layer. Decoder consists of repetitions of a convolution, batch-normalization, and an up-sampling layer. Typically, CED compresses the features along the encoder, and then reconstructs the features along the decoder. In our problem, the orignal Softmax layer at the last layer is modified to a convolution layer, to make CED fully convolutional network.

### 3.2. Redundant CED Network (R-CED)

Here, we propose an alternative convolutional network architecture, namely Redundant Convolutional Encoder-Decoder (R-CED) network. R-CED consists of repetitions of a convolution, batch-normalization, and a ReLU activation layer (see **Fig.**3, each block represents a feature). No pooling layer is present, and thus no up-sampling layer is required. As opposite to CED, R-CED encodes the features into higher dimension along the encoder and achieves compression along the decoder. The number of filters are kept symmetric: at the encoder, the number of filters are gradually increased, and at the decoder, the number of filters are gradually decreased. The last layer is a convolution layer, which makes R-CED a fully convolutional network.

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

### 3.3. Bypass Connections

For CED, R-CED, and CR-CED, bypass connections are added to the network to facilitate the optimization in the training phase and improve performance. Between two different bypass schemes — skip connections in [14] and residual connections in [15] — we chose to use skip connections in [14] which is more suitable for symmetric encoder-decoder design. Bypass connections are illustrated in **Fig.**2 and **Fig.**3 as an 'addition' operation symbol with an arrow. Bypass connections are added every other layer.

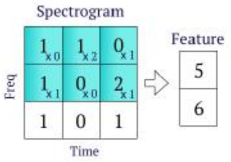

**Fig. 4**: 1-d Convolution

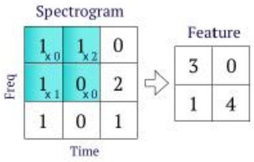

**Fig. 5**: 2-d Convolution

### 3.4. 1-Dim Convolution Operation for Convolution Layers

At all convolution layers throughout the paper, convolution was performed only in 1 direction (see **Fig.**4). In **Fig.**4, the input (3 × 3 white matrix) and the filter (2 × 3 blue matrix) has the same dimension in time axis, and convolution is performed in frequency axis. We found this more efficient than 2-dim convolution (see **Fig.**5) for our input spectra (129 × 8).

## 4. EXPERIMENTAL METHODS

### 4.1. Preprocessing

**Dataset:** The experiment was conducted on the TIMIT database [19] and 27 different types of noise clips were collected from freely available online resource [20]. The noise are mostly babble, but includes different types of noise like 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. After all feature transformation steps were completed, 20% of the training features were assigned as the validation set.

**Feature Transformation:** The audio signals were down-sampled to 8kHz, and the silent frames were removed from the signal. The spectral vectors were computed using a 256-point Short Time Fourier Transform (32ms hamming window) with a window shift of 64-point (8ms). The frequency resolution was 31.25 Hz (=4kHz/128) per each frequency bin. 256-point STFT magnitude vectors were reduced to 129-point by removing the symmetric half. For FNN/RNN, the input feature consisted of a noisy STFT magnitude vector (size: 129×1, duration: 32ms). For CNN, the input feature consisted of 8 consecutive noisy STFT magnitude vectors

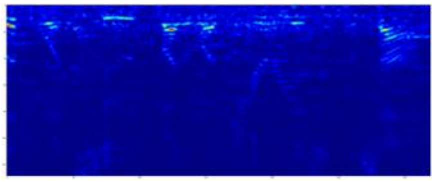

**Fig. 6**: Noisy Spectrogram

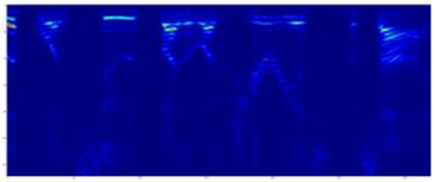

**Fig. 7**: Clean Spectrogram

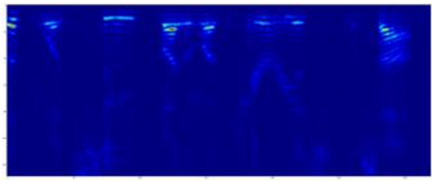

**Fig. 8**: Denoised Spectrogram

(size: $129 \times 8$, duration: 100ms). Both input features were standardized to have zero mean and and unit variance.

**Phase Aware Scaling:** To avoid extreme differences (more than 45 degree) between the noisy and clean phase, the clean spectral magnitude was encoded as similar to [21]:

$$\mathbf{s}_{\text{phase aware}} = \mathbf{s}_{\text{clean}} \cos(\theta_{\text{clean}} - \theta_{\text{noisy}}).$$

Besides, spectral phase was not used in the training phase. At reconstruction, noisy spectral phase was used instead to perform inverse STFT and recover human speech. However, because human ear is not susceptible to phase difference smaller than 45 degree, the distortion was negligible. Through 'phase aware scaling', the phase mismatch be smaller than 45 degree, and For all networks, the output feature consisted of a 'phase aware' magnitude vector (size: $129 \times 1$, duration: 32ms), and were standardized to have zero mean and and unit variance.

### 4.2. Optimization

Fully connected and convolution layer weights were initialized as in [22] and recurrent layer weights were initialized as in [23]. Fully connected and recurrent layer weights were pretrained from the networks of smaller depth with same number of nodes. Convolution layers were trained from scratch, with the aid of batch normalization layer [17] added after each convolution layer [1]. All networks were trained using back propagation with gradient descent optimization using Adam [24] 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 didn't decrease for more than 4 epochs, learning rate was decreased to $lr/2, lr/3, lr/4$, subsequently. The training was repeated once more for FNN and RNN with $\ell_2$ regularization ($\lambda = 10^{-5}$) which slightly improved the performance.

### 4.3. Evaluation Metric

Signal to Distortion Ration (SDR) [25] was used to measure the amount of $\ell_2$ error present between clean and denoised speech:

$$SDR := 10 \log_{10} \frac{||\mathbf{y}||^2}{||f(\mathbf{x}) - \mathbf{y}||^2}.$$

SDR is inversely associated with the objective function presented in (1). In addition, Short time Objective Intelligibility (STOI) [26] and Perceptual Evaluation of Speech Distortion (PESQ) [27] — both assume that human perception has short term memory and hence the error is measured nonlinearly in time of interest— were used to measure the subjective quality of listening.

## 5. EXPERIMENTAL SETUP

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

The first experiment compared CNN with FNN and RNN to demonstrate how feasible it is to use CNN for speech enhancement. Network configurations (e.g. number of nodes, number of layers) that yielded the best performance for each network are summarized in **Table.**2. The best FNN and RNN architectures have 4 fully connected (FC) layers whereas CNN has 16 convolutional layers.

### 5.2. Test 2: CED vs. R-CED

In the second experiment, R-CED was compared to CED. For fair comparison, the total number of parameters were fixed to 33,000 (roughly 132MB of memory) while the depth of the network was fixed to 10 convolution layers. The filter width per each layer is determined accordingly such that i) the symmetric encoder-decoder structure is maintained, ii) the number of parameters are gradually increased and decreased, iii) the 'frequency coverage' is equal for both network. Here, 'frequency coverage' refers to how much nearby frequency bins at the input are used to reconstruct a single frequency bin at the output. We made sure that both networks utilize the same amount of frequency bins to reconstruct a single frequency bin. The configurations for Test 2 is summarized at top two rows of **Table.**1.

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

In the third experiment, we tested how far the performance can be improved using the R-CED network. The R-CED and CR-CED network with skip connections of various network size and depth are compared. The network size (the number of parameters) considered are 33K (132MB memory) and 100K (400MB memory). The depth of the network considered are 10, 16, and 20 convolution layers. Bottom 3 rows in **Table.**1 summarize the network configurations for Test 3.

| | # of Layers | # of Nodes |
|---|---|---|
| **FNN** | (FC, ReLU) × 3,<br>FC | 1024-1024-1024 |
| **RNN** | (Recurrent, ReLU) × 3,<br>FC | 256-256-256 |
| **CNN**<br>(R-CED) | (Conv, ReLU, BN) × 15,<br>Conv | 10-12-14-15-19-21-23-25<br>23-21-19-15-14-12-10-1 |

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

[1] We note that adding BN layers on both FNN and RNN did not improve convergence nor performance for this experiment.

## 6. RESULTS

### 6.1. Test 1: FNN vs. RNN vs. CNN

**Fig.**9 illustrates the denoising performance of FNN, RNN and CNN (left), and the corresponding network size (=the number of parameters, right). All networks exhibited similar performance based on both subjective (STOI, PESQ) and objective quality (SDR) measure. On the other hand, the model size of CNN was about 68 times smaller than that of FNN and about 12 times smaller than RNN. We note that FNN and RNN were optimized to have the smallest network architectures. This experiment validates that CNN requires far lesser number of parameters per layer due to its weight sharing property, and yet can achieve similar or better performance compared to FNN and RNN. 33,000 parameters for CNN are roughly 132MB of memory which can be implemented in an embedded system. Refer to **Fig.**6, **Fig.**7, **Fig.**8 for the example of noisy spectrogram, clean spectrogram, and denoised spectrogram from CNN respectively.

### 6.2. Test 2: CED vs. R-CED

The denoising performance of CED and R-CED are shown in **Fig.**10 with first 4 bars. The R-CED with skip connections showed the best performance, whereas the CED without skip connections showed the worst performance. Regardless of the presence of skip connections, R-CED yielded better results than CED.

The effect of the skip connection was prominent in CED (5.96 to 7.92). This implies that the decoder itself could not reconstruct the ‘lost’ information compressed at the encoder, unless the ‘lost’ information was provided by the skip connections. In addition, the resulting speech from CED sounded artificial and mechanical. This confirms that the decoder could not reconstruct what is necessary for audios to sound like human speech.

On the other hand, the effect of the skip connection was not that notable in R-CED (8.07 to 8.19). This is because R-CED rather expands than compresses the input at the encoder, which can be viewed as mapping a spectrum to higher dimension (e.g. kernel method). By generating redundant representations of important features at the encoder, and removing unwanted features at the decoder, the speech quality was effectively enhanced.

### 6.3. Test 3: Finding the Best R-CED Network

The denoising performance of R-CED of various network size and depth are presented in **Fig.**10 with the last five bars. A few interesting observations are i) the network size 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 when other conditions were kept same (16 convolution layers, 33K parameters).

## 7. CONCLUSION

In this paper, we aimed to find a memory efficient denoising method that can be implemented in an embedded system. Inspired by past success of FNN and RNN, we hypothesized that CNN can effectively denoise speech with smaller network size according to its weight sharing property. We set up an experiment to denoise human speech from babble noise which is a major discomfort to the users of hearing aids. Through experiments, we demonstrated that CNN can yield similar or better performance with much lesser number of model parameters compared to FNN and RNN. Also, we proposed a new fully convolutional network architecture R-CED, and showed its efficacy in speech enhancement. 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 applied to other interesting domains. Our future work will include pruning the R-CED to minimize the operation count of convolution.

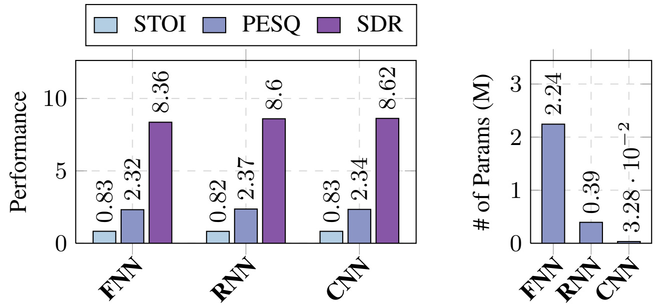

**Fig. 9**: A Comparison of Denoising Performance and the Network Size for FNN vs. RNN vs. CNN

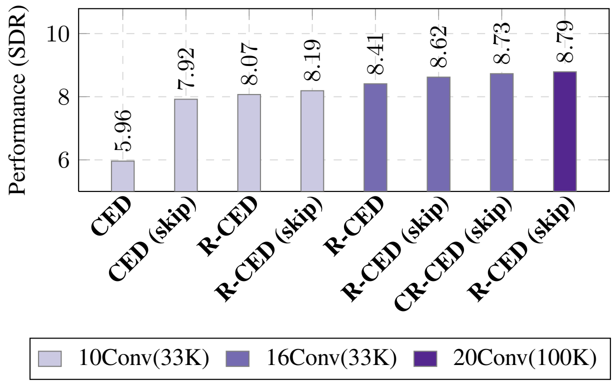

**Fig. 10**: A Comparison of Denoising Performance and the Model Size for different CNN Archtectures

## 8. REFERENCES

[1] Steven 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] Jae S Lim and Alan V Oppenheim, “Enhancement and bandwidth compression of noisy speech,” *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[3] Yariv Ephraim and David 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] Pascal 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*. IEEE, 1996, vol. 2, pp. 629–632.

[5] Yariv Ephraim and Harry 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] Nitish Krishnamurthy and John HL Hansen, "Babble noise: modeling, analysis, and applications," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 7, pp. 1394–1407, 2009.

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

[8] Kun Han, Yuxuan Wang, and DeLiang 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] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui 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] Bingyin Xia and Changchun Bao, "Speech enhancement with weighted denoising auto-encoder.," in *INTERSPEECH*, 2013, pp. 3444–3448.

[11] Keiichi Osako, Rita Singh, and Bhiksha 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] Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong 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] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[14] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang, "Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections," *arXiv preprint arXiv:1603.09056*, 2016.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[16] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine 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.

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

[18] Vinod Nair and Geoffrey 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.

[19] John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David 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] Vincent Akkermans, Frederic Font, Jordi Funollet, Bram De Jong, Gerard Roma, Stelios Togias, and Xavier 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] Pejman Mowlaee and Rahim Saeidi, "Iterative closed-loop phase-aware single-channel speech enhancement," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1235–1239, 2013.

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

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

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[25] Emmanuel Vincent, Rémi Gribonval, and Cédric 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] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech.," in *ICASSP*, 2010, pp. 4214–4217.

[27] Antony W Rix, John G Beerends, Michael P Hollier, and Andries 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*. IEEE, 2001, vol. 2, pp. 749–752.

| | **Layer Configuration** | **Number of Filters** | **Filter Width** |
|---|---|---|---|
| **CED, CED /w Bypass**<br>(11 Conv) | Encoder: (Conv, BN, ReLU, Pool) × 5,<br>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 Bypass**<br>(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 Bypass**<br>(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 |

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