DENSELY CONNECTED NEURAL NETWORK WITH DILATED CONVOLUTIONS FOR REAL-TIME SPEECH ENHANCEMENT IN THE TIME DOMAIN

Ashutosh Pandey[1] *and DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{pandey.99, wang.77}@osu.edu

ABSTRACT

In this work, we propose a fully convolutional neural network for real-time speech enhancement in the time domain. The proposed network is an encoder-decoder based architecture with skip connections. The layers in the encoder and the decoder are followed by densely connected blocks comprising of dilated and causal convolutions. The dilated convolutions help in context aggregation at different resolutions. The causal convolutions are used to avoid information flow from future frames, hence making the network suitable for real-time applications. We also propose to use sub-pixel convolutional layers in the decoder for upsampling. Further, the model is trained using a loss function with two components; a time-domain loss and a frequency-domain loss. The proposed loss function outperforms the time-domain loss. Experimental results show that the proposed model significantly outperforms other real-time state-of-the-art models in terms of objective intelligibility and quality scores.

Index Terms— time domain, fully convolutional, dense network, time-frequency loss, speaker- and noise-independent

1. INTRODUCTION

Speech enhancement is concerned with improving the intelligibility and quality of a speech signal corrupted by additive noise. It is used as a preprocessor in many applications such as automatic speech recognition, telecommunication, hearing aids, and cochlear implants.

In recent years, speech enhancement has been formulated as a supervised learning problem and deep neural networks have been extensively explored [1]. Supervised approaches to speech enhancement generally convert the speech signal to a time-frequency (T-F) representation, and a target signal constructed from the T-F representation is used as the training target. The most popular training targets are ideal ratio mask (IRM) [2], phase-sensitive mask (PSM) [3], and short-time Fourier transform (STFT) magnitude. These training targets are utilized to enhance only the STFT magnitude. The mixture phase is used unaltered for the time-domain signal reconstruction.

The phase of noisy speech is not enhanced mainly due to no clear learnable structure in it [4] and was believed to be unimportant for speech enhancement [5]. A more recent study demonstrated that the phase is important for the perceptual quality of speech, especially at low signal-to-noise ratio (SNR) conditions [6]. This has led researchers to explore algorithms to enhance both the phase and the magnitude using deep neural networks.

The two popular approaches to enhance both the phase and the magnitude using deep learning are complex-domain enhancement and time-domain enhancement. In complex enhancement, generally, a DNN is trained to map the noisy STFT to the complex IRM or the clean STFT. It has been explored in [4, 7, 8, 9, 10] with promising results. The time-domain approaches do not require the frequency-domain transformation where models are trained to directly predict the clean raw samples from the noisy samples. Additionally, the time domain networks can learn to extract features or representations that are well suited for the particular task of speech enhancement. Representative time-domain methods include [11, 12, 13].

In this work, we propose a fully convolutional neural network for real-time speech enhancement in the time domain. The proposed network is an encoder-decoder based architecture with skip connections. Our novel contribution is to add densely connected blocks [14] with dilated convolutions after each layer in the encoder and the decoder. Additionally, we employ sub-pixel convolutional layers instead of transposed convolutions for upsampling. The dilated and densely connected blocks help in long-range context aggregation over different resolutions of the signal. We also propose to train the model using a loss that is a combination of a time-domain loss and a frequency-domain loss.

The rest of this paper is organized as follows. We describe the proposed approach in Section 2. The experimental setup and results are given in Section 3. Section 4 concludes this paper.

This research was supported in part by two NIDCD (R01 DC012048 and R01 DC015521) grants and the Ohio Supercomputer Center.

2. MODEL DESCRIPTION

2.1. Dilated convolutions

Dilated convolutions are used to increase the receptive field of a convolutional neural network and are becoming increasingly popular as an efficient alternative to long short-term memory networks (LSTMs) for learning long-range dependencies. In a dilated convolution with a dilation rate of r, $r-1$ zeros are inserted between the consecutive coefficients of a filter. A dilation rate of r in a filter of size M, increases the receptive field from M to $(M-1)*(r-1)+M$. The receptive field can be set to arbitrarily large size by using an exponentially increasing dilation rate within the network. General practice is to use a dilation rate sequence of form $\{1, 2, 4, 8, 16, ...\}$. The dense block in our model comprises of dilated and causal convolutions. The causal convolutions are used across the frames to make sure that there is no leakage of information from the future frames. Note that we do not use causal convolutions within a frame. An illustrative diagram of dilated and causal convolution is shown in Fig. 1.

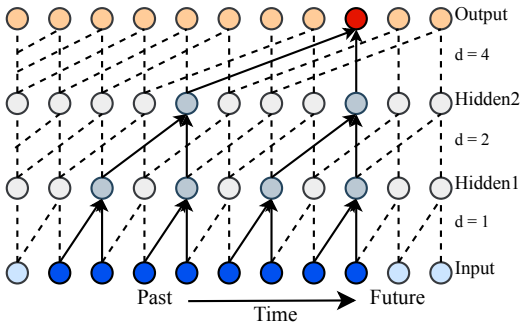

Fig. 1: An example of dilated causal convolution with a filter of size 2.

2.2. Densely connected networks

Densely connected networks (DCN) were recently proposed in [14]. In a DCN, the inputs to a given layer in the network are a concatenation of the outputs from all the previous layers. This approach has two major advantages. First, the dense connection to all the previous layers avoids the vanishing gradient problem. Second, a thinner network is empirically found to outperform a much wider network and hence improving the parameter efficiency of the network. In our model, we propose a dilated dense block that is used after each layer in the encoder and the decoder of the model. An illustrative diagram of the proposed dense block is shown in Fig. 2.

Each dense block consists of five layers of 2-dimensional convolutions. The convolutions across the frames are causal. The causal convolutions make sure that the proposed approach is suitable for real-time implementation. Each convolution is followed by layer normalization [15] and parametric ReLU (PReLU) nonlinearity [16]. The dilation rates in each dense block are set to 1, 2, 4, 8 and 16 as shown in Fig 2.

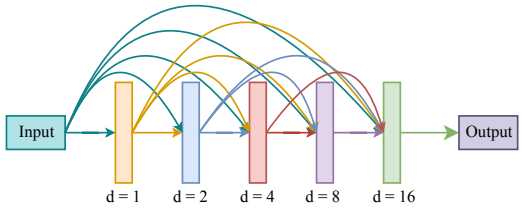

Fig. 2: The proposed dilated dense block. The dilation rate is exponentially increased from 1 to 16.

2.3. Sub-pixel convolutions

Sub-pixel convolutions are used as a learnable upsampling layer within a convolutional neural network. It was proposed in [17] for image super-resolution. In this work, we use sub-pixel convolution as a better alternative for transposed convolution in the decoder, to avoid the checkerboard artifacts [18]. In a transposed convolution, the input signal is first upsampled by inserting zeros between the consecutive samples followed by a convolutional layer to get a signal with non-zero entries. This leads to having an asymmetric configuration if the stride is not divisible by the filter length causing the checkerboard artifacts [17]. In sub-pixel convolution, the convolution is performed over the original signal (without inserting zeros) and the output number of channels is increased by a multiplicative factor of the upsampling rate. The extra channels are reshaped to get the desired upsampled signal. An illustrative diagram of the upsampling of a 1D signal by a factor of 2 using sub-pixel convolution is shown in Fig 3.

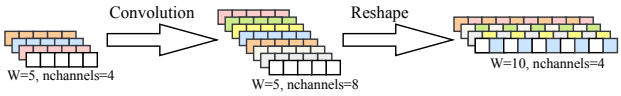

Fig. 3: An illustration of sub-pixel convolution for upsampling of a 1D signal by a factor of 2.

2.4. Model architecture

The schematic diagram of the model architecture is shown in the left part of Fig 4. The model consists of an input layer, an encoder, dilated and dense blocks, a decoder, and an output layer. All the convolutions except at the output layer follow layer normalization and PReLU. The input to the model is of size $[batch\_size, 1, num\_frames, frame\_size]$. The input layer uses filters of size $(1, 1)$ to increase the number of channels to 64. The input layer is followed by a dense block. The convolutions in all the dense blocks use filters of size $(2, 3)$ with 64 output channels. Each layer in the encoder first halves the dimension (downsampling) along the frame axis (last axis) using a convolution with a stride of $(1, 2)$ and filters of size $(1, 3)$. The downsampling is followed by a dense

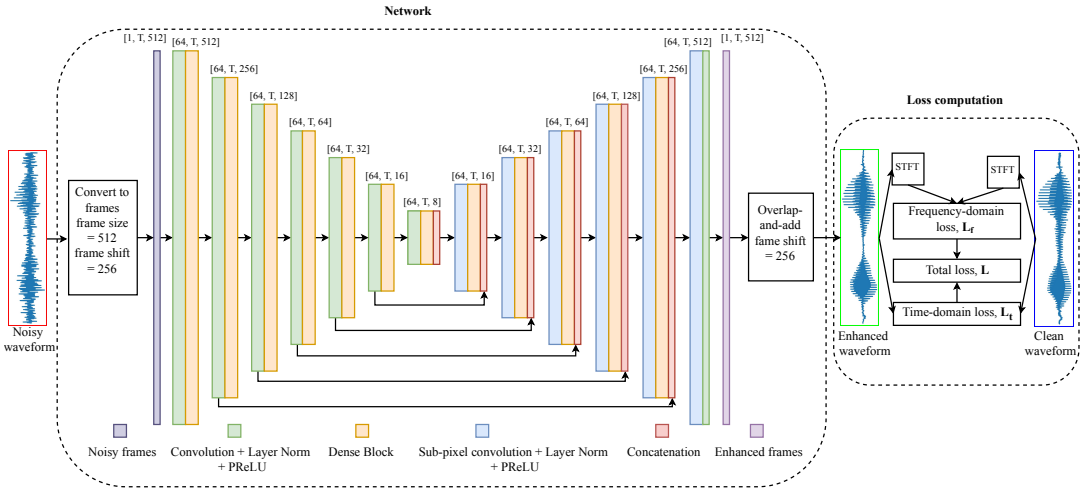

Fig. 4: The proposed model and loss function.

block. The dense blocks after each layer in the encoder help in context aggregation at different resolutions. There are six such layers in the encoder and the final output of the encoder is of size $[batch\_size, 64, num\_frames, frame\_size/64]$.

The decoder uses sub-pixel convolutions and dense blocks to successively reconstruct the signal to the original size. The input to each layer in the decoder is a concatenation (along the channel axis) of the previous layer output and the output from the corresponding symmetric layer in the encoder. The sub-pixel convolutions use filters of size $(1, 3)$ to double the input size along the frame axis. Finally, the output layer uses filters of size $(1, 1)$ to output the enhanced frames with one channel.

2.5. Loss function

We use a combination of two losses for the model training. First, the enhanced frames are converted to a waveform using overlap-and-add method. An utterance level loss is computed in the time domain using mean squared error between the enhanced utterance and the clean utterance. The time-domain loss is defined as:

$$L_t(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{M} \sum_{n=0}^{M-1} (x_i[n] - \hat{x}_i[n])^2 \qquad (1)$$

where $x[n]$ and $\hat{x[n]}$ denote the n^{th} sample of the clean and the enhanced utterance respectively and M is utterance length.

Second, we take STFT of the utterances and use L_1 loss [19] over the L_1 norm of the STFT coefficients as in [20, 11]. The frequency-domain loss is given by:

$$L_f(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{T \cdot F} \sum_{t=1}^{T} \sum_{f=0}^{F} |(|X(t,f)_r| + |X(t,f)_i|) - (|\hat{X}(t,f)_r| + |\hat{X}(t,f)_i|)| \qquad (2)$$

where $X(t, f)$ and $\hat{X}(t, f)$ are the TF bins of STFTs of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ respectively. T in the number of frames and F is the number of frequency components. x_r and x_i denote the real and imaginary part of complex variable x.

Finally, the time and frequency domain loss are combined in following way.

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \alpha * L_t(\boldsymbol{x}, \hat{\boldsymbol{x}}) + (1 - \alpha) * L_f(\boldsymbol{x}, \hat{\boldsymbol{x}}) \qquad (3)$$

where α is a hyper-parameter that is tuned on the validation set. An illustrative diagram of the loss computation is shown in the right part of Fig 4.

3. EXPERIMENTS

3.1. Datasets

We evaluate our system in a speaker- and noise-independent way by training it on a large number of noises and speakers. We use 7138 utterances from the WSJ0 SI-84 dataset [21]. It consists of 83 speakers (42 males and 41 females) in which 76 are used for training and remaining 6 (3 males and 3 females) are used for evaluation.

For training, we use 10000 non-speech sounds from a sound effect library (available at `www.sound-ideas.com`) [22] and generate 320000 utterances at the SNRs of -5dB, -4dB, -3dB, -2dB, -1dB and 0dB. A noisy utterance is created in the following way. First, an utterance from the training speakers, an SNR, and a noise type are randomly selected. Then the selected utterance is mixed with a random segment of the selected noise type at the selected SNR.

For the test set, we use two noises (babble and cafeteria) from an Auditec CD (available at `http://www.auditec.com`), and generate 150 mixtures at each SNR of -5dB, -2dB, 0dB, 2dB, and 5dB. For the validation set, we use 6 speakers from the training set (150 utterances) and mix it with factory noise at an SNR of -5 dB.

Table 1: Model comparisons in terms of STOI and PESQ scores on untrained speakers.

	test noise	ADTBabble						ADTCafeteria					
	test SNR	-5 dB	-2 dB	0 dB	2 dB	5 dB	Average	-5 dB	-2 dB	0 dB	2 dB	5 dB	Average
STOI (%)	Mixture	58.42	65.53	70.52	75.02	81.30	70.16	57.05	64.71	69.65	74.46	80.95	69.36
	CRN	80.30	86.82	89.61	91.50	93.61	88.37	78.10	85.10	88.24	90.61	92.99	87.01
	AECNN-SM	81.48	88.25	91.06	92.96	94.81	89.71	79.54	87.00	89.84	92.02	94.11	88.50
	TCNN	82.80	88.90	91.25	92.98	94.75	90.14	80.60	87.10	89.81	91.94	94.01	88.69
	DDAEC-T	83.48	89.53	91.92	93.55	95.23	90.74	81.38	87.78	90.53	92.50	94.54	89.35
	DDAEC-TF	**84.03**	**90.29**	**92.53**	**94.15**	**95.67**	**91.33**	**82.08**	**88.50**	**91.13**	**93.01**	**94.97**	**89.94**
PESQ	Mixture	1.56	1.71	1.82	1.94	2.12	1.83	1.46	1.63	1.77	1.91	2.12	1.78
	CRN	2.18	2.50	2.67	2.82	3.01	2.64	2.17	2.46	2.63	2.78	2.97	2.60
	AECNN-SM	2.21	2.60	2.80	2.97	3.17	2.75	2.23	2.60	2.76	2.93	3.12	2.73
	TCNN	2.18	2.52	2.70	2.86	3.06	2.66	2.14	2.45	2.62	2.78	2.98	2.59
	DDAEC-T	2.23	2.57	2.75	2.91	3.12	2.72	2.21	2.51	2.70	2.86	3.07	2.67
	DDAEC-TF	**2.30**	**2.71**	**2.91**	**3.08**	**3.28**	**2.86**	**2.32**	**2.65**	**2.83**	**2.99**	**3.20**	**2.80**

3.2. Baselines

For the baselines, we train 4 different models. First, we train a complex spectrogram mapping based model proposed in [8]. We call this model CRN in our comparisons. Second, we train a time-domain model that is a frame-based system, with large frame size (1.024 seconds), trained using a loss over STFT magnitudes [20]. We call this model AECNN-SM. Finally, we train the TCNN model proposed in [13].

3.3. Experimental settings

All the utterances are resampled to 16 kHz. The frames are extracted using a rectangular window of size 32 ms and an overlap of 16 ms. In each epoch of the training, we chunk a random segment of 4 seconds from an utterance if it is larger than 4 seconds. The smaller utterances are zero-padded to match the size of the largest utterance in the batch. The Adam optimizer is used for stochastic gradient descent (SGD) based optimization. We train the model for 15 epochs with a batch size of 4 utterances and a learning rate schedule given in Table 2. At the training time, we observe the short-time objective intelligibility (STOI) [23] score on the validation set after each epoch of training and the model with the maximum STOI is used for evaluation. We set α in Equation 3 to 0.8.

We provide the code for our implementation at `https://github.com/ashutosh620/DDAEC`.

Table 2: Learning rate schedule for training the proposed model.

Epochs	1 to 3	4 to 9	10 to 12	13 to 15
Learning rate	0.0002	0.0001	0.00005	0.00001

3.4. Experimental results

We compare all the models in terms of STOI whose values typically range between 0 and 1 and perceptual evaluation of speech quality (PESQ) whose values range from -0.5 to 4.5 [24]. The results are given in Table 2. We call the proposed model DDAEC standing for dilated and dense autoencoder. We report two results on DDAEC, one trained only using time-domain loss (DDAEC-T) and the other trained using proposed time-frequency loss (DDAEC-TF).

First, we observe that the DDAEC-T model outperforms all the baseline models in terms of STOI. For PESQ, it outperforms all the baseline models except AECNN-SM which is a frame-based model with large frame size, hence not suitable for real-time implementation. But, when using time-frequency loss, the DDAEC outperforms all the baseline models for both scores at all SNR conditions. For STOI, the best baseline is TCNN and an average improvement of 1.19% and 1.24% is obtained for babble and cafeteria noise respectively. For PESQ, the best baseline is AECNN-SM and improvements of 0.11 and 0.17 are obtained for the two noises. The proposed DDAEC-T and DDAEC-TF significantly outperform the CRN model which is a frequency domain model for complex-spectrogram mapping. This demonstrates the superiority of a time-domain model over a frequency domain model. Similarly, DDAEC-T and DDAEC-TF both outperform another time-domain model TCNN.

Next, we compare the proposed model in terms of the number of parameters as listed in Table 3. The proposed model has the fewest number of parameters followed by TCNN. The CRN has the maximum number of parameters even though it uses group LSTMs to reduce the number of parameters.

Table 3: Model comparisons in terms of number of trainable parameters.

Model	CRN	AECNN-SM	TCNN	DDAEC
# of parameters in millions	9.06	6.45	5.8	4.82

4. CONCLUSIONS

We have proposed a novel fully convolutional neural network for speech enhancement. The model utilizes dense connections with dilated convolutions for long-range context aggregation. The proposed model is suitable for real-time implementation and outperforms the other state-of-the-art models in terms of objective intelligibility and quality scores. Future work includes exploring the proposed model for multi-channel speech enhancement and other speech preprocessing tasks such as speaker separation and speech dereverberation.

5. REFERENCES

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1702–1726, 2018.

[2] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[3] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *ICASSP*, 2015, pp. 708–712.

[4] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 3, pp. 483–492, 2016.

[5] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[6] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *speech communication*, vol. 53, no. 4, pp. 465–494, 2011.

[7] S.-W. Fu, T.-y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[8] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *ICASSP*, 2019, pp. 6865–6869.

[9] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex U-Net," *arXiv preprint arXiv:1903.03107*, 2019.

[10] A. Pandey and D. Wang, "Exploring deep complex networks for complex spectrogram enhancement," in *ICASSP*, 2019, pp. 6885–6889.

[11] ——, "A new framework for supervised speech enhancement in the time domain," in *INTERSPEECH*, 2018, pp. 1136–1140.

[12] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 9, pp. 1570–1584, 2018.

[13] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP*, 2019, pp. 6875–6879.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[15] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[17] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[18] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[19] A. Pandey and D. Wang, "On adversarial training and loss functions for speech enhancement," in *ICASSP*, 2018, pp. 5414–5418.

[20] ——, "A new framework for cnn-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 27, no. 7, pp. 1179–1188, 2019.

[21] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Workshop on Speech and Natural Language*, 1992, pp. 357–362.

[22] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2604–2612, 2016.

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[24] 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 *ICASSP*, 2001, pp. 749–752.