TCNN: TEMPORAL CONVOLUTIONAL NEURAL NETWORK 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

This work proposes a fully convolutional neural network (CNN) for real-time speech enhancement in the time domain. The proposed CNN is an encoder-decoder based architecture with an additional temporal convolutional module (TCM) inserted between the encoder and the decoder. We call this architecture a Temporal Convolutional Neural Network (TCNN). The encoder in the TCNN creates a low dimensional representation of a noisy input frame. The TCM uses causal and dilated convolutional layers to utilize the encoder output of the current and previous frames. The decoder uses the TCM output to reconstruct the enhanced frame. The proposed model is trained in a speaker- and noise-independent way. Experimental results demonstrate that the proposed model gives consistently better enhancement results than a state-of-the-art real-time convolutional recurrent model. Moreover, since the model is fully convolutional, it has much fewer trainable parameters than earlier models.

***Index Terms*—** noise-independent and speaker-independent speech enhancement, real-time implementation, time domain, temporal convolutional neural network, TCNN

1. INTRODUCTION

Speech enhancement is the task of removing or attenuating additive noise from a speech signal. It is used as a preprocessor in many applications such as robust speech recognition, teleconferencing and hearing aids. Traditional speech enhancement approaches include spectral subtraction methods [1], Wiener filtering [2], statistical model-based methods [3] and nonnegative matrix factorization [4].

In the past few years, deep learning based supervised methods have become the mainstream for speech enhancement [5]. Generally, in a supervised approach, a given speech signal is converted to a time-frequency (T-F) representation, and a target signal constructed from the T-F representation is used as the training target. Some of the most popular training targets are ideal ratio mask (IRM) [6], phase sensitive mask (PSM) [7] and short-time Fourier transform (STFT) magnitude.

Even though using the T-F representation is the most popular approach, it has some disadvantages. First, these methods generally ignore the clean phase information and use the noisy phase for the time domain signal reconstruction. Some studies in the past have demonstrated that the phase is necessary for better speech quality, especially in low signal-to-noise ratio (SNR) conditions [8]. Second, some of the training targets, such as the IRM, do not lead to perfect signal reconstruction even when an ideal target is used. Finally, for fast speech enhancement, the computation of the T-F representation is an additional overhead.

The factors mentioned above and the powerful representation capability of deep neural networks (DNNs) have led researchers to explore DNNs for speech enhancement in the time domain. In [9], the authors demonstrate the effectiveness of fully convolutional neural networks for time domain speech enhancement. Recently in [10], the authors train a model employed in the time domain with a frequency domain loss to improve the perceptual quality of the enhanced speech. Even though the work in [10] can obtain state-of-the-art performance, it does not address the problem of real-time enhancement. The proposed model uses a 128 ms frame at each time step making the model unsuitable for real-world applications.

Motivated by the successful implementation of the TCNNs for sequence modeling [11], and the effectiveness of encoder-decoder based architecture for the time domain speech enhancement [10, 12], we propose to combine both of them to obtain a real-time enhancement system. The proposed model has an encoder-decoder based architecture that consists of causal convolutional layers. A TCM is inserted between the encoder and the decoder to learn the long-range dependencies from the past. The TCM used in our work is similar to the one used in [13] where the authors use it to perform real-time speaker separation in the time domain with state-of-the-art performance.

This paper is organized as follows: We first describe the TCNNs in the next section. Section 3 describes the proposed framework. Experimental details, results, and comparisons are given in Section 4. Section 5 concludes the paper.

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

2. TEMPORAL CONVOLUTIONAL NEURAL NETWORKS

TCNNs are generic convolutional networks proposed for sequence modeling tasks with causal constraint [11]. Given an input sequence $x_0, ..., x_t$ and the corresponding output sequence $y_0, ..., y_t$, a sequence modeling network learns to estimate the output sequence $\hat{y}_0, ..., \hat{y}_t$ by training the network on some loss function between the estimated sequence and the output sequence. The causal constraint on the network implies that the prediction $\hat{y}_t$ depends only on the $x_0, ..., x_t$ but not on the future inputs $x_{t+1} ..., x_T$. In the case of speech enhancement in the time domain, the input sequence is the sequence of noisy frames, and the output sequence is the sequence of clean frames.

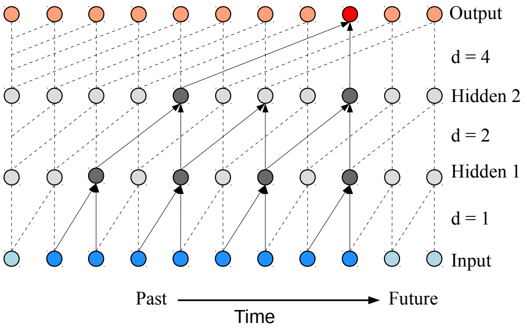

Fig. 1. An example of dilated causal convolution with a filter of size 2.

To impose the causal constraint, the TCNNs are comprised of causal and dilated convolutional layers. The causal convolutions ensure that there is no leakage of information from the future to the past. The dilated convolutions help to increase the receptive field. The larger the receptive field, the more a network can look into the past. Fig. 1 illustrates the example of a dilated and causal convolution with a filter of size 2.

Additionally, a TCNN is comprised of residual blocks so that a deep network can be adequately trained using residual learning [14]. Fig. 2 shows the residual block used in this work. A similar residual block has been used in [13]. The residual block consists of 3 convolutions: input 1x1 convolution, depthwise convolution, and output 1x1 convolution. The input convolution is used to double the number of incoming channels. The output convolution is used to get back to the original number of channels, which makes the addition of the inputs and outputs compatible. The depthwise convolution is used to reduce the number of parameters further. In a depthwise convolution, the number of channels is kept the same, and only one filter per input channel is used for the output computation [15]. In a normal convolution, each output channel uses as many filters as the number of channels in the input. The input and the middle convolutions are followed by parametric ReLU non-linearity [16] and batch normalization [17].

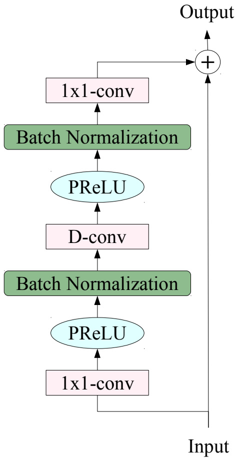

Fig. 2. The residual block used in the proposed framework.

3. PROPOSED FRAMEWORK

The proposed TCNN has three components: the encoder, the decoder, and the TCM. The encoder and the decoder are comprised of two-dimensional causal convolutional layers while the TCM consists of one-dimensional causal and dilated convolutional layers. A block diagram of the proposed framework is shown in Fig. 3.

The encoder takes the sequence of noisy frames as input. The size of the input to the encoder is $T \times 320 \times 1$, where T is the number of frames, 320 is the frame size, and 1 is the number of input channels. The first layer in the encoder increases the number of channels from 1 to 64. The output dimension after the first layer is $T \times 320 \times 16$. The next seven layers successively reduce the size along the frame dimension using convolutions with a stride of two along that dimension. The final output of the encoder is of dimension $T \times\ 4\ \times 64$. None of the layers in the network modifies the size along the time dimension so that the output has the same number of frames as in the input. Each layer in the encoder is followed by batch normalization and parametric ReLU non-linearity.

The output of the encoder is reshaped to a one-dimensional signal of size $T \times 256$. The TCM operates on the reshaped output and produces an output of the same size. The TCM has three dilation blocks stacked together. A dilation block is formed by stacking six residual blocks having exponentially increasing dilation rates. In a dilation block, the successive dilation rates in the residual blocks are 1, 2, 4, 8, 16 and 32.

The decoder is a mirror-image of the encoder and consists of a series of two-dimensional causal transposed convolutional (deconvolutional) layers. The output of the decoder after each layer is concatenated with the outputs from the corresponding symmetric layer in the encoder. At the training time, we add a dropout of 0.3 to the incoming skip connections from the encoder. Each layer in the decoder is followed by batch normalization and parametric ReLU non-linearity.

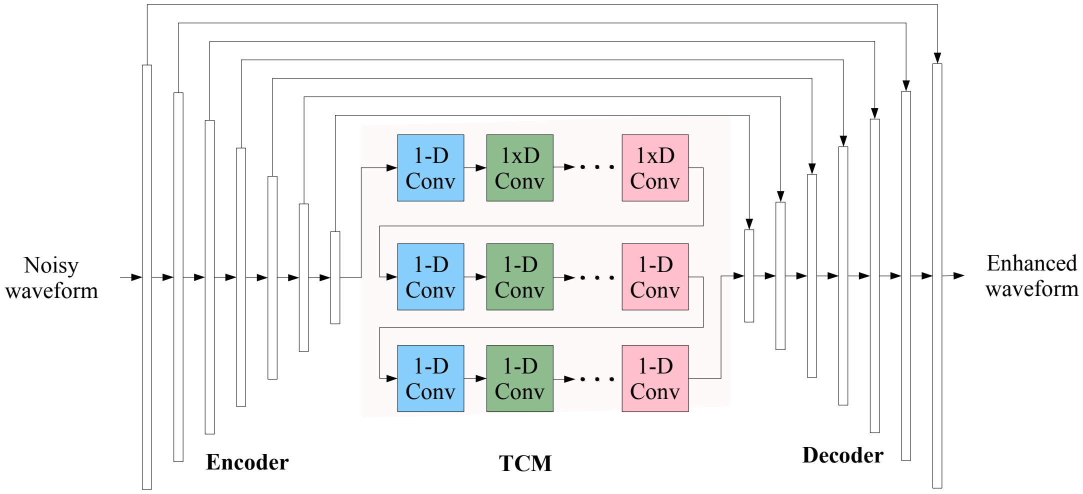

Fig. 3. The proposed TCNN model.

Detailed network parameters are given in Table 1. For the encoder and the decoder, the hyperparameters are in the format *filterHeight* × *filterWidth, (stride along time, stride along frame)*. For the TCM, the entries enclosed by the small braces represent a residual block, and the hyperparameters are in the format *filterSize, dilationRate, outputChannels*.

Table 1. The proposed model architecture. T denotes the number of time frames. The residual blocks are shown in brackets.

layer name	input size	hyperparameters	output size
reshape_1	$T \times 320$	-	$T \times 320 \times 1$
conv2d_1	$T \times 320 \times 1$	$2 \times 5, (1, 1)$	$T \times 320 \times 16$
conv2d_2	$T \times 320 \times 16$	$2 \times 5, (1, 2)$	$T \times 160 \times 16$
conv2d_3	$T \times 160 \times 16$	$2 \times 5, (1, 2)$	$T \times 79 \times 16$
conv2d_4	$T \times 79 \times 16$	$2 \times 5, (1, 2)$	$T \times 39 \times 32$
conv2d_5	$T \times 39 \times 32$	$2 \times 5, (1, 2)$	$T \times 19 \times 32$
conv2d_6	$T \times 19 \times 32$	$2 \times 5, (1, 2)$	$T \times 9 \times 64$
conv2d_7	$T \times 9 \times 64$	$2 \times 5, (1, 2)$	$T \times 4 \times 64$
reshape_2	$T \times 4 \times 64$	-	$T \times 256$
TCM	$T \times 256$	$\left.\begin{array}{c} \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{1}, 512 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{2}, 512 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{4}, 512 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{8}, 512 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{16}, 512 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 512 \\ 3, \mathbf{32}, 512 \\ 1, 1, 256 \end{pmatrix} \end{array}\right\} \times 3$	$T \times 256$
reshape_3	$T \times 256$	-	$T \times 4 \times 64$
deconv2d_7	$T \times 4 \times 128$	$2 \times 5, (1, 2)$	$T \times 9 \times 64$
deconv2d_6	$T \times 9 \times 128$	$2 \times 5, (1, 2)$	$T \times 19 \times 32$
deconv2d_5	$T \times 19 \times 64$	$2 \times 5, (1, 2)$	$T \times 39 \times 32$
deconv2d_4	$T \times 39 \times 64$	$2 \times 5, (1, 2)$	$T \times 79 \times 16$
deconv2d_3	$T \times 79 \times 32$	$2 \times 5, (1, 2)$	$T \times 160 \times 16$
deconv2d_2	$T \times 160 \times 32$	$2 \times 5, (1, 2)$	$T \times 320 \times 16$
deconv2d_1	$T \times 320 \times 16$	$2 \times 5, (1, 1)$	$T \times 320 \times 1$
reshape_4	$T \times 320 \times 1$	-	$T \times 320$

4. EXPERIMENTS

4.1. Datasets

We evaluate the proposed framework in a speaker- and noise-independent way on the WSJ0 SI-84 dataset [18]. The WSJ0 SI-84 dataset consists of 7138 utterances of 83 speakers (42 males and 41 females). We select six speakers for the test set. The remaining seventy-seven speakers are used to create training mixtures. For training noises, we use 10000 non-speech sounds from a sound effect library (available at www.sound-ideas.com). The training utterances are generated at the SNRs of -5 dB, -4 dB, -3 dB, -2 dB, -1 dB and 0 dB. 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. In total, 320000 training utterances are generated. The duration of the training noises is around 125 hours, and that of the training utterances is around 500 hours.

For the test set, we use two challenging noises (babble and cafeteria) from an Auditec CD (available at http://www.audit-ec.com). Two test sets are created. The first test set uses the utterances of 6 speakers (3 males and 3 females) from the training speakers. The second test set is created from the utterances of 6 (3 males and 3 females) speakers that are not included in the training set. The two test sets assess the performance on trained and untrained speakers. Note that all test utterances are excluded from the training set.

4.2. Baselines

For the baselines, we train two models. First, we train an LSTM based real-time causal system. We call this model LSTM in our results. From the input layer to the output layer, the LSTM model has 161, 1024, 1024, 1024, 1024, and 161

Table 2. Model comparisons in terms of STOI and PESQ scores on trained speakers.

evaluation metrics	STOI (%)						PESQ					
test SNR	-5 dB			-2 dB			-5 dB			-2 dB		
noises	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.
unprocessed	58.9	57.4	58.2	66.3	65.2	65.8	1.63	1.52	1.58	1.79	1.70	1.75
LSTM	77.3	74.3	75.8	82.6	81.4	82.0	2.06	2.04	2.05	2.36	2.30	2.33
CRN	79.7	76.1	77.9	85.5	82.7	84.1	2.17	2.12	2.15	2.44	2.38	2.41
TCNN	**83.3**	**80.5**	**81.9**	**89.2**	**86.9**	**88.1**	**2.22**	**2.15**	**2.19**	**2.56**	**2.44**	**2.50**

Table 3. Model comparisons in terms of STOI and PESQ scores on untrained speakers.

evaluation metrics	STOI (%)						PESQ					
test SNR	-5 dB			-2 dB			-5 dB			-2 dB		
noises	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.
unprocessed	58.5	57.2	57.9	65.4	64.7	65.1	1.56	1.47	1.52	1.69	1.63	1.66
LSTM	75.2	73.4	74.3	82.7	80.8	81.8	1.94	1.97	1.96	2.26	2.24	2.25
CRN	78.0	74.8	76.4	84.4	82.2	83.3	2.04	2.03	2.04	2.34	2.31	2.33
TCNN	**82.8**	**80.6**	**81.7**	**88.9**	**87.1**	**88.0**	**2.18**	**2.14**	**2.20**	**2.52**	**2.45**	**2.50**

Table 4. Model comparisons in terms of number of trainable parameters.

Model	Number of of parameters in millions
LSTM	36.81
CRN	17.6
TCNN	**5.10**

units. Second, we train another real-time causal system recently proposed in [19]. This system is a recurrent convolutional architecture that uses an encoder-decoder based convolutional network with LSTMs for recurrence. We call this model CRN in our results. Note that both the baseline models operate in the frequency domain.

4.3. Experimental settings

All the utterances are resampled to 16 kHz. The frames are extracted using a rectangular window of size 20 ms and overlap of 10 ms. All the models are trained using mean squared error loss and a batch size of 8 utterances. The small utterances are zero padded to match the size of the largest utterance in the batch. The Adam optimizer [20] is used for stochastic gradient descent (SGD) based optimization. The learning rate is set to a small constant value equal to 0.0002.

4.4. Experimental results

We compare the models in terms of short-term objective intelligibility (STOI) [21] and perceptual evaluation of speech quality (PESQ) [22] scores. First, we compare the TCNN with the baselines on trained speakers. The results are given in Table 2. When compared with LSTM, an average improvement of 6.1 % is observed in STOI on both the SNRs. PESQ is improved by 0.14 on -5 dB and 0.17 on -2 dB. Similarly, when compared with the CRN, the STOI is improved by 4 % on both the SNRs and PESQ is improved by 0.04 on -5 dB and 0.09 on -2 dB.

Next, we compare the models on untrained speakers. The results are given in Table 3. A similar trend is observed in the performance improvement except that in this case, the TCNN also significantly outperforms the CRN for PESQ scores. This indicates that the CRN model overfits for the speakers in the training set.

We also compare the number of trainable parameters in the models. The numbers are given in Table 4. The proposed model has much fewer parameters when compared with the baseline models, making it suitable for the efficient implementation in real-world applications.

Finally, it is worth mentioning that the proposed framework can accept a variable frame size at the input. The only required change is to either add or remove layers from the encoder and the decoder depending on the desired frame size. Furthermore, this model can be easily applied to other regression-based supervised speech processing tasks such as speaker separation, dereverberation, and echo cancellation.

5. CONCLUSIONS

In this study, we have proposed a novel, fully convolutional neural network for real-time speech enhancement in the time domain. The proposed TCNN significantly outperforms existing real-time systems in the frequency domain. Additionally, the proposed framework has much fewer trainable parameters. Furthermore, the system is easy to adapt to a different frame size by simple modifications in the encoder and the decoder of the network. Future research includes exploration of the TCNN model for other speech processing tasks such as dereverberation, echo cancellation and speaker separation.

6. ACKNOWLEDGEMENTS

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

7. 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] P. Scalart et al., "Speech enhancement based on a priori signal to noise estimation," in *Proceedings of ICASSP*, 1996, vol. 2, pp. 629–632.

[3] P. C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Boca Raton, FL, USA, 2nd edition, 2013.

[4] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[5] 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.

[6] 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.

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

[8] K. Paliwal, K. Wojcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465 – 494, 2011.

[9] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," *arXiv preprint arXiv:1703.02205*, 2017.

[10] A. Pandey and D. Wang, "A new framework for supervised speech enhancement in the time domain," in *Proceedings of Interspeech*, 2018, pp. 1136–1140.

[11] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[12] S. Pascual, A. Bonafonte, and J. Serr, "SEGAN: Speech enhancement generative adversarial network," in *Proceedings of Interspeech*, 2017, pp. 3642–3646.

[13] Y. Luo and N. Mesgarani, "TasNet: Surpassing ideal time-frequency masking for speech separation," *arXiv preprint arXiv:1809.07454*, 2018.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, pp. 1610–02357, 2017.

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

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

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

[19] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proceedings of Interspeech*, 2018, pp. 3229–3233.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] 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.

[22] 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 *Proceedings of ICASSP*, 2001, pp. 749–752.