FILTER-AND-CONVOLVE: A CNN BASED MULTICHANNEL COMPLEX CONCATENATION ACOUSTIC MODEL

Peidong Wang[1] *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
{wang.7642, wang.77}@osu.edu

ABSTRACT

We propose a convolutional neural network (CNN) based multichannel complex-domain concatenation acoustic model. The proposed model extracts speech-specific information from multichannel noisy speech signals. In addition, we design two CNN templates that have wide applicability and several speaker adaptation methods for the multichannel complex concatenation acoustic model. Even with a simple BeamformIt beamformer and the baseline language model, our method obtains a word error rate (WER) of 5.39% on the CHiME-4 corpus, outperforming the previous best result by 13.06% relatively. Using an MVDR beamformer, our model outperforms the corresponding best system by 9.77% relatively.

Index Terms— convolutional neural network, multichannel acoustic model, concatenation, complex, CHiME-4

1. INTRODUCTION

In multichannel speech recognition tasks, multiple signals are recorded simultaneously. The spatial information from multiple recordings is typically used in the form of a spatial filter by beamformers. The speech-specific information contained in the multiple signals, however, may be insufficiently exploited by spatial filtering.

Beamformers are used to steer the microphone array to the direction of the target speaker, hence attenuating the noises coming from other directions [1, 2]. Commonly used beamformers for speech recognition include weighted delay-and-sum beamformers such as BeamformIt [3], minimum variance distortionless response (MVDR) beamformers [4], and generalized eigenvalue (GEV) beamformers [5]. BeamformIt aligns the signals based on generalized cross-correlation with phase-transform (GCC-PHAT) values. It then assigns weights to and sums the signals. A typical MVDR beamformer works in the frequency domain. It tries to minimize the noise power while keeping the speech power unchanged in the steered direction. To avoid an explicit estimation of the direction-of-arrival (DOA), the GEV beamformer is proposed. It maximizes the signal-to-noise ratio (SNR) for each output frequency bin separately.

Different from beamforming, prior work on multichannel concatenation acoustic modeling has attempted to make full use of the speech-specific information in multiple channels, but at the cost of not benefiting from explicit spatial information. Liu *et al.* concatenate the perceptual linear prediction (PLP) features and feed them to a deep neural network (DNN) based acoustic model [6]. Swietojanski *et al.* compare beamforming and concatenation for convolutional neural network (CNN) based acoustic models [7]. Their result, however, indicates that concatenating 40-dimensional log Mel features as the input to the acoustic model performs worse than using a beamformer frontend.

Recently, DNN based time-frequency masking has been demonstrated to be very effective for the accurate estimation of the steering vector and power spectral density (PSD) matrices [8, 9, 10]. This can be viewed as a way of incorporating the speech related information into the beamformer. In this work, we propose a multichannel concatenation acoustic model that extracts the speech-specific information after the noisy signals are spatially filtered. Consistent improvements are observed over the conventional monaural systems. Using BeamformIt and an MVDR beamformer as frontends, our models outperform previous best systems by 13.06% and 9.77% relatively.

The rest of this paper is organized as follows. In Section 2, we describe our CNN based multichannel complex concatenation acoustic model. In Sections 3 and 4, the experimental setup and evaluation results are presented. Finally, we make some concluding remarks in Section 5.

2. SYSTEM DESCRIPTION

In a filter-and-sum beamformer, the spatially filtered signals are combined using the summation operation. In our filter-and-convolve system, we substitute the summation operation in the beamformer with a trainable convolutional layer in the acoustic model. We will show the reason of this substitution, the framework of our system, the templates for the convolutional layer, and the speaker adaptation methods for our system in the following sections.

2.1. Convolution as a Substitute for Summation

A typical 2-D convolution operation with one output channel can be expressed as the formula below.

$$y_{ij} = \sum_{c=0}^{C-1} (\sum_{a=0}^{T-1} \sum_{b=0}^{T-1} w_{ab}^{c} x_{(i+a)(j+b)}^{c}) + b_{ij} \tag{1}$$

where y_{ij} denotes element i, j of the output channel, w_{ab}^{c} element a, b in the $T \times T$ template of channel c, $x_{(i+a)(j+b)}^{c}$ input element $(i + a), (j + b)$ of channel c, and b_{ij} the bias for element i, j of the output channel; the total number of input channels is denoted as C.

If we set the sizes of the templates to 1×1's, $w^{c} = 1$, and $b_{ij} = 0$, the convolution operation can be simplified to the summation below.

$$y_{ij} = \sum_{c=0}^{C-1} x_{ij}^{c} \tag{2}$$

This shows that summation may be viewed as a special case of convolution. If not overfitted, trainable CNN based concatenation models may be as good as, if not better than, summation based ones.

2.2. Model Design

2.2.1. Multichannel Complex Concatenation Acoustic Model

A diagram showing the whole system is in Fig. 1. The multichannel signals are fed into the beamformer to get spatially filtered. The filtered signals are then preprocessed and transformed to the frequency domain separately. After concatenation, the frequency-domain features are used as the input to our CNN based multichannel complex concatenation acoustic model, comprising a convolutional layer, conventional feature extraction modules, and an internal monaural acoustic model. Note that the preprocessing and short-time Fourier transform (STFT) module may be omitted if the beamformer directly outputs frequency-domain features.

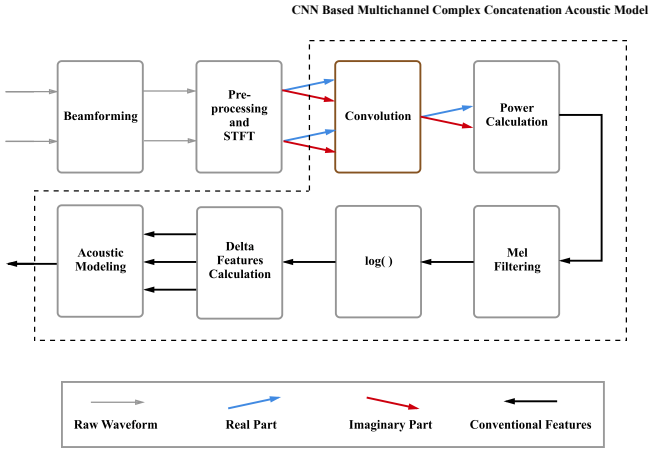


Fig. 1. The Diagram of the CNN Based Multichannel Complex Concatenation Acoustic Model

Although both taking multichannel signals directly as the input, our model differs from the joint *beamformer - acoustic model* systems. Xiao *et al.* propose to jointly train the acoustic model with a neural network predicting beamforming weights from GCC values [11]. Heymann *et al.*'s *Beamnet* jointly trains the acoustic model with the PSD mask estimation network in the GEV beamformer [12]. Hoshen *et al.* propose to use a multichannel convolution over time (tConv) layer to replace the beamformer and the Mel feature extractor [13]. Note that tConv is different from the convolution in CNN. Sainath *et al.* conduct extensive work on joint multichannel separation and acoustic modeling using raw waveforms [14]. In addition to a single tConv layer for both beamforming and feature extraction, they also propose a factored version separating the two operations. Adaptive variants reestimating a subset of the spatial filter coefficients based on the input are also proposed [15, 14].

Most of the joint *beamformer - acoustic model* systems sum the spatially filtered signals directly. This is the main difference from our system, which uses a trainable convolutional layer to extract the speech-specific information. The second tConv operation in the factored version of Sainath *et al.*'s system, although does not do summation, differs from our system since the input is from the model-specific "look directions" and the convolution templates are shared among different channels.

2.2.2. CNN Templates

We design two kinds of templates for the convolutional layer, the weighted CNN templates and the complex CNN templates. The weighted CNN method shares the templates for the real and imaginary parts of each channel, while the complex CNN templates are designed such that the convolution operation of each channel mimics a complex multiplication.

Figures showing the two kinds of templates are in Fig. 2. The real part of the output is denoted as o_r, the imaginary part as o_i, the real part of channel c as r_c, and the imaginary part as i_c. For the weighted CNN method, the template for channel c is denoted as t_c. For the complex CNN method, two templates, denoted as t_{cr} and t_{ci}, are used for channel c.

For a detailed illustration, we denote the contribution of channel c to the real part of the output as o_{cr} and to the imaginary part of the output as o_{ci}. If the biases are set to zeros, we have the formulas below.

$$o_r = \sum_{c=0}^{C-1} o_{cr} \tag{3}$$

$$o_i = \sum_{c=0}^{C-1} o_{ci} \tag{4}$$

For simplicity, let us take the example when the sizes of the templates are 1×1's. The weighted CNN method can be expressed as the formulas below.

$$o_{cr} = r_c * t_c \tag{5}$$

$$o_{ci} = i_c * t_c \tag{6}$$

The two formulas can be rewritten as the formula below.

$$\begin{aligned} & o_{cr} + j o_{ci} \\ =& r_c * t_c + j i_c * t_c \\ =& (r_c + j i_c) * t_c \end{aligned} \tag{7}$$

The corresponding formulas for the complex CNN method are as follow.

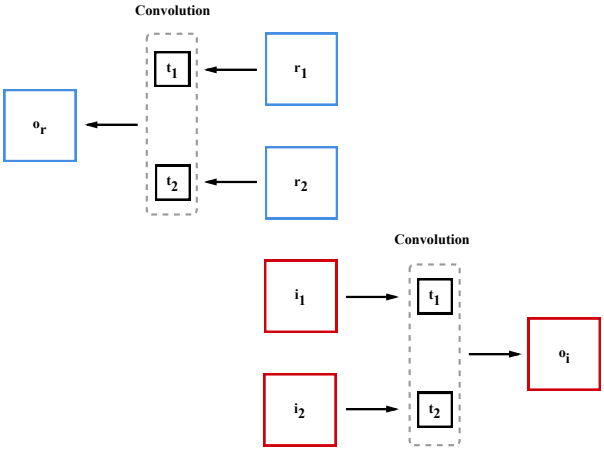
$$o_{cr} = r_c * t_{cr} - i_c * t_{ci} \tag{8}$$

$$o_{ci} = r_c * t_{ci} + i_c * t_{cr} \tag{9}$$

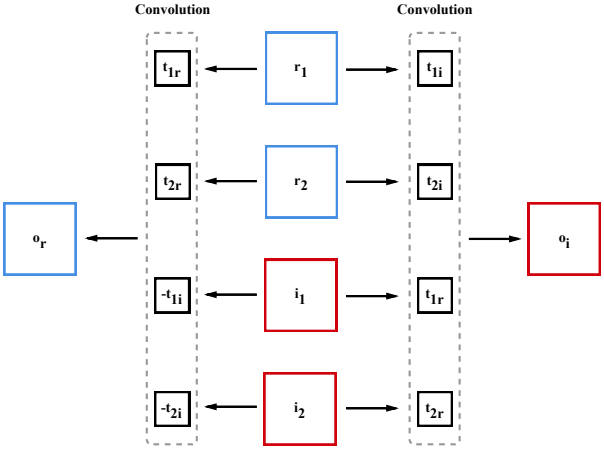
The two formulas above can be viewed as the complex multiplication of $r_c + j i_c$ and $t_{cr} + j t_{ci}$, as in the formula below.

$$\begin{aligned} & o_{cr} + j o_{ci} \\ =& (r_c * t_{cr} - i_c * t_{ci}) + j(r_c * t_{ci} + i_c * t_{cr}) \\ =& (r_c + j i_c) * (t_{cr} + j t_{ci}) \end{aligned} \tag{10}$$

For the two kinds of templates, the weighted CNN templates may be viewed as special cases of the complex CNN ones.



(a) weighted CNN



(b) complex CNN

Fig. 2. Templates for the Convolutional Layer

2.3. Unsupervised Speaker Adaptation

We use four unsupervised speaker adaptation methods for our model. Three of the methods are based on the linear input network (LIN) and the fourth one the linear hidden network (LHN) [16, 17].

The input to multichannel complex concatenation acoustic models differs from that to conventional monaural models in two aspects, the multiple channels and the real and imaginary parts of each channel. We design three LIN based template sharing methods accordingly, shared templates, different templates for different channels, and different templates for both different channels and different complex components.

The LHN based method adds a linear layer at the input of the internal monaural acoustic model, mimicking the LIN adaptation for conventional monaural acoustic models.

3. EXPERIMENTAL SETUP

3.1. Dataset

Our experiments are conducted on the 4th CHiME speech separation and recognition challenge (CHiME-4) corpus. It is a read speech corpus with a target of distant-talking automatic speech recognition. There are two types of data, real recorded and artificially simulated. The real data is recorded in real noisy environments, including bus, cafe, pedestrian area, and street junction. The simulated data, on the other hand, is generated by artificially mixing clean speech with noisy backgrounds.

The training set of the CHiME-4 corpus contains 1600 real utterances and 7318 simulated utterances for each of the six microphone channels. The real utterances are uttered by 4 speakers and the simulated utterances are from the 83 speakers of the WSJ0 SI-84 training set. The development set consists of 410*6 real utterances and 410*6 simulated utterances for each of the four audio environments. Similarly, the test set has 330*6 real recordings and 330*6 simulated utterances for each environment. 12% of the real data are influenced by hardware issues or masked by the user's hands or clothes.

3.2. Implementation Details

3.2.1. Monaural Acoustic Model

The monaural acoustic model in this work is an improved version of the wide residual bidirectional long short-term memory (BLSTM) network (WRBN) [18, 19]. We apply an utterance-wise recurrent dropout method to fine tune the original WRBN. The dropout masks for the hidden vectors are sampled once per utterance. The input gate, output gate, forget gate, and the cell update vector use four different masks. Using Adam optimizer with an initial learning rate of 10^{-5}, we fine tune the WRBN for five epochs. This fine-tuned monaural acoustic model is used as the initial model in all the following experiments.

3.2.2. BeamformIt as Frontend

We use the BeamformIt script provided in the Kaldi toolkit [20]. All the settings remain default, except that $do\_indiv\_channels$ is set to 1 to output individual signals before summation. Note that the selection of channels also follows the default in the script, i.e. only channel 1, 3, 4, 5, and 6 are used.

The five spatially filtered waveforms are preprocessed separately by dithering, removing the direct currency (dc) offset, conducting pre-emphasis, and applying the *povey* window [20]. The output signals are then converted to the frequency domain and concatenated.

To train the weighted CNN model, we fine tune the convolutional layer together with the internal monaural acoustic model. The sizes of the templates are 3×3's. The elements at the centers are initialized to ones, and all the other elements zeros. The optimizer is Adam and the initial learning rate is set to 10^{-8} to prevent overfitting. In addition, we also set the dropout rate to 0.1 during the fine-tuning process.

The complex CNN model is trained in a slightly different way. We reuse the internal acoustic model of the best weighted CNN model, keep it fixed, and fine tune the complex CNN layer only. The sizes of the templates are also 3×3's. For the initial values of channel c, t_{cr} has a one at the center but t_{ci} does not. All the non-one elements are initialized to random values ranging from -0.01 to 0.01. All the other settings are the same as the weighted CNN model.

3.2.3. MVDR Beamformer as Frontend

Our MVDR beamformer uses a feedforward neural network to estimate the PSD mask [8]. The input to the neural network is 19

consecutive frames of the utterance-wise mean-normalized spectrum features. The model has four hidden layers, with 2048 nodes in each layer. The hidden activation functions are exponential linear units (elus) and the output is compressed by the sigmoid function. The median of the masks, i.e. the median of the network output values, is used as the mask for the PSD estimation.

The model is trained for 30 epochs with an Adagrad optimizer and a dropout rate of 0.1. Note that for both the MVDR beamformer and the GEV beamformer below, we use all the six microphone channels.

3.2.4. GEV Beamformer as Frontend

We use the publicly available GEV beamformer provided by Heymann *et al.* [9]. The PSD mask estimation is based on the provided BLSTM model. We also apply the blind analytic normalization (BAN) in the beamformer. Note that this version of the GEV beamformer is weaker than the one used in Heymann *et al.*'s CHiME-4 system [19].

4. EVALUATION RESULTS

4.1. Results and Comparisons

With BeamformIt, the MVDR beamformer and the GEV beamformer as frontends, our models yield WERs of 5.39%, 3.97%, and 4.14%, respectively. Comparisons between our results (rounded) and the previous best results are shown in Table 1.

We use the final result of Dat *et al.*'s submission as the best result using BeamformIt [21]. Their *BeamformIt-I2Rb*, yielding a WER of 6.44%, may be the best system using BeamformIt as the frontend. For the best system using an MVDR frontend, we choose Erdogan *et al.*'s submission [22]. For the two results better than Erdogan *et al.*'s, Du *et al.* apply an iterative GEV-based generalized sidelobe canceller and Heymann *et al.* use the GEV beamformer [23, 24, 19]. Since our method does not involve system ensembling, we use Heymann *et al.*'s result as the best result using the GEV beamformer [19].

Table 1. WER (%) Comparisons With the Previous Best Systems

Beamformer	BeamformIt	MVDR	GEV
Previous Best [21, 22, 19]	6.2	4.4	**3.9**
Proposed	**5.4**	**4.0**	4.1

Our system using BeamformIt and the baseline language model outperforms the previous best system by 13.06% relatively. Given the simplicity of the BeamformIt based system, the 5.39% WER may be considerably good. Using the MVDR beamformer, our system outperforms the corresponding best system by 9.77% relatively. For the GEV based systems, the performance difference is mainly caused by the beamformer itself. We can easily verify this by comparing the baseline (*Beamformed*) result 4.87% in Table 2 with the corresponding result 4.07% in Heymann *et al.*'s paper [19].

4.2. Step-by-Step Results

The results after each step is shown in Table 2. *Beamformed* denotes the baseline models using beamformers and the monaural acoustic model. The row *Concatenated* contains the results using the untrained multichannel concatenation model.

Table 2. Step-by-Step WERs (%)

Beamformer	BeamformIt	MVDR	GEV
Beamformed	6.31	4.29	4.87
Concatenated	6.25	4.25	4.90
weighted CNN	6.14	4.23	4.75
complex CNN	**6.10**	**4.20**	**4.73**

The WERs of the BeamformIt and MVDR beamformer based systems are reduced by simply concatenating the multichannel signals. For the GEV based system, the WER slightly increases after the concatenation, possibly due to the distortion introduced in the GEV beamforming process [5]. The complex CNN systems are better than the *Beamformed* ones by 3.33%, 2.10%, and 2.87% relatively. The consistent improvements indicate that our model is widely applicable to different frontends.

4.3. Speaker Adaptation Results

The results using different speaker adaptation methods are shown in Table 3. *Shared LIN*, *Channel LIN*, and *Channel+Complex LIN* denote shared templates, different templates for different channels, and different templates for both different channels and different complex components, respectively.

Table 3. WERs (%) Using Different Adaptation Methods

Adaptation	BeamformIt	MVDR	GEV
Shared LIN	5.97	4.19	4.67
Channel LIN	6.07	4.18	4.74
Channel+Complex LIN	6.07	4.15	4.67
LHN	**5.39**	**3.97**	**4.14**

The LHN based method performs the best among the four speaker adaptation methods. The reason may be that the nonlinear feature extraction operations in our model may make the features more speaker-invariant.

5. CONCLUDING REMARKS

We propose a CNN based multichannel complex concatenation acoustic model. It exploits not only the spatial information but also the speech-specific information in the multiple signals. In addition to a backend applicable to various beamformer frontends, our model may also be viewed as a step towards the multichannel acoustic model that implicitly embeds the beamformer using a network adaptive to different utterances. Other future directions include better designs of the speech feature extraction layer and the corresponding templates, new speaker adaptation methods for complex concatenation models, the application of our model in a frequency-wise manner as a post-filter, and the application of our model to end-to-end systems.

6. ACKNOWLEDGMENTS

We would like to thank Z.Q. Wang for sharing the MVDR code and K. Tan for valuable comments on an early version of this paper. This work was supported in part by an AFRL contract (FA8750-15-1-0279), an NSF grant (IIS-1409431), and the Ohio Supercomputer Center.

7. REFERENCES

[1] D.L. Wang and J. Chen, "Supervised speech separation based on deep learning: an overview," *arXiv preprint arXiv:1708.07524*, 2017.

[2] Z. Zhang, J. Geiger, J. Pohjalainen, A.E. Mousa, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *arXiv preprint arXiv:1705.10874*, 2017.

[3] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[4] O.L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[5] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[6] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5542–5546.

[7] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[8] X. Zhang, Z.Q. Wang, and D.L. Wang, "A speech enhancement algorithm by iterating single-and multi-microphone processing and its application to robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 276–280.

[9] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 196–200.

[10] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust mvdr beamforming using time-frequency masks for online/offline asr in noise," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5210–5214.

[11] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M.L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5745–5749.

[12] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5325–5329.

[13] Y. Hoshen, R.J. Weiss, and K.W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4624–4628.

[14] T.N. Sainath, R.J. Weiss, K.W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, et al., "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[15] B. Li, T.N. Sainath, R.J. Weiss, K.W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition.," in *INTERSPEECH*, 2016, pp. 1976–1980.

[16] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[17] B. Li and K.C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[18] P. Wang and D.L. Wang, "Utterance-wise recurrent dropout and iterative speaker adaptation for robust monaural speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, 2018.

[19] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," *submitted to the CHiME*, vol. 4, 2016.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[21] T.H. Dat, N.W.Z. Terence, S. Sivadas, L.T. Tuan, and T.A. Dung, "The i2r system for chime-4 challenge," *in Proceedings of CHiME-4 Workshop*, 2016.

[22] H. Erdogan, T. Hayashi, J.R. Hershey, T. Hori, C. Hori, W.N. Hsu, S. Kim, J. Le Roux, Z. Meng, and S. Watanabe, "Multi-channel speech recognition: Lstms all the way through," in *CHiME-4 workshop*, 2016.

[23] J. Du, Y.H. Tu, L. Sun, F. Ma, H.K. Wang, J. Pan, C. Liu, J.D. Chen, and C.H. Lee, "The ustc-iflytek system for chime-4 challenge," *Proc. CHiME*, pp. 36–38, 2016.

[24] A. Krueger, E. Warsitz, and R. Haeb-Umbach, "Speech enhancement with a gsc-like structure employing eigenvector-based transfer function ratios estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 206–219, 2011.