INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Extract, Adapt and Recognize: an End-to-end Neural Network for Corrupted Monaural Speech Recognition

Max W. Y. Lam[1,2,*], *Jun Wang*[1], *Xunying Liu*[2], *Helen Meng*[2], *Dan Su*[1], *Dong Yu*[3]

[1]Tencent AI Lab, Shenzhen, China
[2]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong SAR, China
[3]Tencent AI Lab, Bellevue WA, USA

{maxwylam, joinerwang, dansu, dyu}@tencent.com

Abstract

Automatic speech recognition (ASR) in challenging conditions, such as in the presence of interfering speakers or music, remains an unsolved problem. This paper presents Extract, Adapt, and Recognize (EAR), an end-to-end neural network that allows fully learnable separation and recognition components towards optimizing the ASR criterion. In between a state-of-the-art speech separation module as an *extractor* and an acoustic modeling module as a *recognizer*, the EAR introduces an *adaptor*, where adapted acoustic features are learned from the separation outputs using a bi-directional long short term memory network trained to minimize the recognition loss directly. Relative to a conventional joint training model, the EAR model can achieve 8.5% to 22.3%, and 1.2% to 26.9% word error rate reductions (WERR), under various dBs of music corruption and speaker interference respectively. With speaker tracing the WERR can be further promoted to 12.4% to 29.0%.

Index Terms: monaural source separation, robust speech recognition, joint training, filter bank learning

1. Introduction

Monaural speech signals containing nonstationary interference such as background music or multi-speaker speech are ubiquitous in real-world circumstances such as vehicle-mounted systems, conference call devices, telephone communications, and online programs. Despite tremendous improvements brought by deep learning technologies [1, 2, 3, 4], conventional automatic speech recognition (ASR) systems could fail in these adverse environments. While some progress has been made for monaural speech separation and recognition [5, 6, 7, 8, 9], it remains one of the most challenging problems in ASR to date.

Existing literature typically divides the problem into speech separation/enhancement and recognition stages [10, 11], allowing modular research to be separately conducted. One straightforward approach in the stage-wise manner is to train the two models separately, and in a testing phase, the speech features enhanced by the separation/enhancement model are then passed to the acoustic model for recognition [12]. However, distortions inevitably introduced by the separation model unseen by the acoustic model would lead to a dramatic drop of performance in such a cascaded framework [13, 14].

To solve this issue, researchers have recently reported a joint training of an acoustic model (AM) and a speech separation (SS) model with considerable error reductions in both noise-robust ASR [11, 10] and multi-speaker ASR [8, 9] tasks. To bridge the SS and AM, some cutting-edge researches [9, 8] introduce self-contained frameworks where the separation is forced to operate directly in a Mel-filter domain to be in line with the AM. However, it is debatable if this may result in a potentially sub-optimal solution for the separation step. Meanwhile, it becomes difficult to integrate fast-evolving third-party SS techniques that are mostly not in the Mel-filter domain. In contrast, other state-of-the-art frameworks typically construct some affine transformation towards Mel-filter bank values (fbanks) through a deep neural network (DNN) network, either to learn the Mel filtering weight matrix [15], or to directly learn the affine transforming function via DNN [16, 17, 11].

These bridging approaches are essentially DNN-based versions, i.e., frame-by-frame affine transforming layers, of the expert-predefined Mel filtering function whose input-output data pairs are a magnitude spectrogram and fbanks. However, the task at hand is much more complicated: the input is imperfect spectrogram that contains spectral estimation errors and temporal distortions; the output to be learned whose objective is defined by the back-propagated criterion from the high-level ASR phoneme classification also majorly adds dynamic complexity to the problem. Indeed for decades before the deep learning era, many research efforts have been devoted [18, 19] to model properly the temporal dynamics in SS problems, and they are remarkably effective. Undoubtedly, it is impossible to represent a complex system with a simple network's capacity.

Inspired by the above, we would like to explore if and to what extent benefits could be brought by designing a more elegant neural network-based architecture. This paper presents Extract, Adapt, and Recognize (EAR), a fully trainable end-to-end neural network. For SS, an *extractor* is modified from a state-of-the-art separation model named deep extractor network (DENet) [20]; for AM, a *recognizer* adopts an advanced network with new recognition training criterion center-loss-optimal convolutional long short-term memory DNN (CL-CLDNN) [21, 22]; in between SS and AM, a novel *adaptor* bridges the separation and recognition components together in an alternative way for incorporating temporal dynamics. To the best of our knowledge, this paper is the first work that reveals the criticality of appropriate network design for adapting the intermediate representations. Section 2 presents the proposed EAR network in more detail. Section 3 shows the evaluation results, and the conclusion is drawn in Section 4.

2. Extract, Adapt and Recognize

The proposed network is illustrated in Figure 1: a) extract target speech's magnitude spectrogram via the DENet, b) adapt

[*]Max did this work while he was an intern at Tencent.

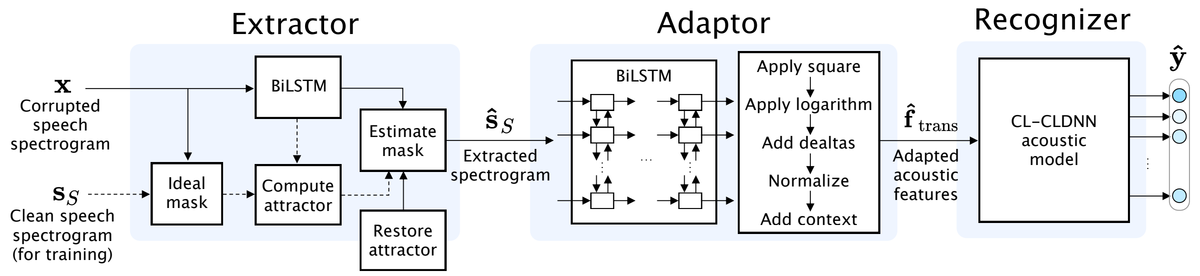

Figure 1: *The schematic diagram of our proposed EAR network*

to learnable acoustic features using a recurrent network, and c) recognize speech by feeding learned features to an AM. End-to-end joint training is applied after each module in the EAR network has been separately pre-trained.

2.1. Extractor

The goal of the *extractor* module in the EAR is to separate the target speech from a monaural mixture signal. A popular method in the field of source separation would be an ideal ratio mask (IRM) [23] method. Its neural network extensions, including the deep attractor network (DANet) [24] and the deep extractor network (DENet) [20], have been proven effective for single-channel speech separation tasks. Aiming at our goal, we present a modified version of DENet with lower complexity.

2.1.1. *Deep extractor network*

In source separation, we consider the observed single-channel mixture signal $x(n)$ as a linear combination of the target speech signal $s_S(n)$ and the interference signal $s_I(n)$: $x(n) = s_S(n) + s_I(n)$. The spectrogram is calculated using short-time discrete Fourier transform (STFT) for the mixture signal and the reference target speech signal, represented as $T \times F$ dimensional features $\mathbf{x}$ and $\mathbf{s}_S$ respectively, where T is the number of frames, and F is the number of frequency bins.

For training, the ideal mask is calculated as $\mathbf{m}_S = |\mathbf{s}_S|/|\mathbf{x}|$. To estimate IRM, a bi-directional LSTM (BiLSTM) [25] network is used to map the mixture signal to a higher-dimensional embedding space $\mathbb{R}^{TF\times K}$ such that the outputs can be reshaped into an embedding matrix:

$$\mathbf{V} = \phi_{\text{BiLSTM}}\left(\mathbf{x}; \boldsymbol{\Theta}_{\text{extract}}\right) \in \mathbb{R}^{TF\times K}, \tag{1}$$

where $\boldsymbol{\Theta}_{\text{extract}}$ denotes the network parameters for $\phi_{\text{BiLSTM}}(\cdot)$.

In our DENet, the main idea is to compute a weighted average of the embeddings belonging to the target speech training samples and store it as the attractor. Thus only one attractor is needed to be calculated in the embedding space:

$$\mathbf{a}_S = \frac{\mathbf{V}^\top(\mathbf{m}_S \odot \mathbf{w})}{\sum\limits_{t=1}^{T}\sum\limits_{f=1}^{F}(\mathbf{m}_S \odot \mathbf{w})}, \tag{2}$$

where $\mathbf{a}_S \in \mathbb{R}^K$, $\odot$ is the element-wise multiplication, and $\mathbf{w} \in \mathbb{R}^{TF}$ is a binary threshold matrix defined as

$$\mathbf{w}_{t,f} = \begin{cases} 0, & \text{if } \mathbf{x}_{t,f} < \max(\mathbf{x})/100 \\ 1, & \text{otherwise} \end{cases}, \tag{3}$$

Then the mask can be estimated for the target speech signal $\hat{\mathbf{m}}_S$ by calculating the similarity between the attractor and the embedding of each T-F bin:

$$\hat{\mathbf{m}}_S = \text{Sigmoid}\left(\mathbf{V}\mathbf{a}_S\right), \tag{4}$$

where the Sigmoid$(\cdot)$ function monotonically scales the estimated mask between $[0,1]$. Finally, the extracted spectrogram of the target speech can be computed by $\hat{\mathbf{s}}_S = \mathbf{x} \odot \hat{\mathbf{m}}_S$. Note that in the training phase the attractors of the target speech training samples are collected, and then the average value of these attractors is calculated, which will be restored as a global average attractor to extract the target speech in the reference phase.

2.2. Adaptor

The *Adaptor* module is an intermediate network that bridges the extraction and recognition components. Its input is the extracted defective spectrogram with both spectral and temporal distortions by the previous module, and its output acoustic features are to be learned for minimizing the recognition loss. In order to properly incorporate the temporal dynamics from both bottom-up and top-down, the *Adaptor* we proposed uses a BiLSTM architecture, which allows efficient estimation of the conditional posterior probability of the complete sequences without making any explicit assumption about their distribution.

In contrast to prior art [15, 16, 17, 11] commonly adopting DNN (often called universal function approximators [26]) with frame-by-frame affine transforming layers representing a Mel-filtering function [27] to predict fbanks, BiLSTM (often called universal program approximators [28]) is able to use all contextual input spectrogram to predict a point in the output acoustic feature space. Actually an optimized output acoustic feature space for *recognizer* is not necessarily a Mel-fbank space, according to our experiment, that we pre-trained an *adaptor* using MSE criterion against clean target fbanks, and then did joint training together with the *recognizer*, but no gain was observed comparing to directly minimizing the recognition loss bypassing pre-train with clean fbanks, and we also observed the finally converged acoustic features drifted away from the target fbanks. Similar observations were also reported in [15] with empirical evidence. Still, here we denote the learned acoustic representations as adaptive fbanks after the naming convention in ASR.

2.2.1. *BiLSTM for adaptive fbank learning*

By feeding the spectrogram into the *adaptor*, we get

$$\hat{\mathbf{f}} = \psi_{\text{BiLSTM}}\left(\hat{\mathbf{s}}_S; \boldsymbol{\Theta}_{\text{adapt}}\right)^2 \in \mathbb{R}_+^{TD}, \tag{5}$$

where D is the number of the filters used, and $\boldsymbol{\Theta}_{\text{adapt}}$ denotes all the parameters needed for $\psi_{\text{BiLSTM}}(\cdot)$. Recall that the standard calculation of fbanks is done by passing the squared spectrogram through the filters; therefore, fbanks must be non-negative. Since the outputs of a standard BiLSTM are unbounded, in the EAR, we apply a square, which performs better than a ReLU according to our evaluations, on the BiLSTM outputs to match the range of fbanks. The estimated fbanks $\hat{\mathbf{f}}$ is then passed through a series of differentiable operations, including element-wise logarithm, adding delta and delta-delta, performing global mean-variance normalization, and splicing the features of a context window of $2W+1$ frames centered at the current frame, to yield the transformed fbanks $\hat{\mathbf{f}}_{\text{trans}} \in \mathbb{R}^{3D(2W+1)}$, where W is the size of the one-sided context window.

2.3. Recognizer

The *recognizer* module in the EAR deals with acoustic modeling, i.e., to classify phonemes based on the adapted acoustic features. To achieve state-of-the-art performance, we adopt an advanced network architecture, namely convolutional LSTM deep neural network (CLDNN) [22], denoted as Center-Loss-optimal CLDNN (CL-CLDNN) [21], as the *recognizer* in the EAR network. The transformed fbanks are fed into the *recognizer* to estimate the phoneme posteriors $\hat{\mathbf{y}}_t$:

$$\mathbf{u} = \boldsymbol{\psi}_{\text{CL-CLDNN}}\left(\hat{\mathbf{f}}_{\text{trans}}; \boldsymbol{\Gamma}\right), \tag{6}$$

$$\hat{\mathbf{y}}_t = \text{Softmax}\left(\mathbf{W}\mathbf{u}_t + \mathbf{b}\right), \tag{7}$$

where $\mathbf{u}_t$ is the second last output of CL-CLDNN at t-th frame, and $\text{Softmax}(\mathbf{z}) = e^{\mathbf{z}}/\|e^{\mathbf{z}}\|_1$ is used as an estimate of the posterior over the phoneme classes. We use $\boldsymbol{\Theta}_{\text{recog}} = \{\boldsymbol{\Gamma}, \mathbf{W}, \mathbf{b}\}$ to denote all the parameters of the *recognizer*.

2.4. Training mechanism

The training of the EAR network consists of two phases: (1) separate pre-training of the extractor, the adaptor and the recognizer, and (2) end-to-end joint training of the whole EAR network.

2.4.1. Pre-training of extractor, adaptor and recognizer

We train the *extractor* by using the typical mean squared error (MSE) loss between the estimated spectrogram and the reference spectrogram of the target speech: $\mathcal{L}_{\text{MSE}} = \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{s}_S^{(i)} - \hat{\mathbf{s}}_S^{(i)}\right\|_2^2$, where M is the batch size of mixture signal samples for training, (i) denotes the index of training examples, and $\|\cdot\|_2$ denotes 2-norm of a vector.

We train the *recognizer* using the Center Loss (CL) [21]:

$$\mathcal{L}_{\text{CL}} = \frac{1}{MT}\sum_{i=1}^{M}\sum_{t=1}^{T} -\mathbf{y}_t^{(i)}\log\hat{\mathbf{y}}_t^{(i)} + \lambda\left\|\mathbf{u}_t^{(i)} - \mathbf{c}_t^{(i)}\right\|_2^2 \tag{8}$$

where λ is a hyperparameter that controls the weighting of CL, $\mathbf{c}_t^{(i)}$ is the class center corresponding to the label at t-th frame, which can be jointly updated with back-propagation. With an incentive to improve the discriminative power of deep features [29], CL learns a center of deep features for each class and penalizes the distance between the deep features and their corresponding class centers.

After the training of the *recognizer* converges, we freeze its parameters and continue to train the *adaptor* using the spectrogram of clean speech as training data, by directly back-propagating the recognition loss $\mathcal{L}_{\text{CL}}$:

$$\frac{\partial\mathcal{L}_{\text{CL}}}{\partial\boldsymbol{\Theta}_{\text{adapt}}} = \frac{\partial\mathcal{L}_{\text{CL}}}{\partial\hat{\mathbf{f}}_{\text{trans}}}\frac{\partial\hat{\mathbf{f}}_{\text{trans}}}{\partial\hat{\mathbf{f}}}\frac{\partial\hat{\mathbf{f}}}{\partial\boldsymbol{\Theta}_{\text{adapt}}} \tag{9}$$

by mini-batch SGD. Note that, directly minimizing the recognition loss is preferred over minimizing MSE between the clean fbanks and the predicted fbanks, because the clean fbanks computed using Mel filtering operations are not necessarily the best features for the AM.

2.4.2. Joint training

After training each module separately as pre-training, we jointly update the parameters in three modules by back-propagating $\mathcal{L}_{\text{CL}}$ all the way to the *extractor*:

$$\frac{\partial\mathcal{L}_{\text{CL}}}{\partial\boldsymbol{\Theta}_{\text{extract}}} = \frac{\partial\mathcal{L}_{\text{CL}}}{\partial\hat{\mathbf{f}}_{\text{trans}}}\frac{\partial\hat{\mathbf{f}}_{\text{trans}}}{\partial\hat{\mathbf{f}}}\frac{\partial\hat{\mathbf{f}}}{\partial\hat{\mathbf{m}}_S}\frac{\partial\hat{\mathbf{m}}_S}{\partial\mathbf{V}}\frac{\partial\mathbf{V}}{\partial\Theta_{\text{extract}}} \tag{10}$$

In practice, the gradient calculation is performed through computational graph-based automatic differentiation [30] in the conventional deep learning programming libraries, e.g., TensorFlow [31]. Subsequently, mini-batch SGD is used to update all parameters in the EAR, $\boldsymbol{\Theta}_{\text{EAR}} = \{\boldsymbol{\Theta}_{\text{extract}}, \boldsymbol{\Theta}_{\text{adapt}}, \boldsymbol{\Theta}_{\text{recog}}\}$ by

$$\boldsymbol{\Theta}_{\text{EAR}} = \boldsymbol{\Theta}_{\text{EAR}} - \alpha\left\{\frac{\partial\mathcal{L}_{\text{CL}}}{\partial\boldsymbol{\Theta}_{\text{extract}}}, \frac{\partial\mathcal{L}_{\text{CL}}}{\partial\boldsymbol{\Theta}_{\text{adapt}}}, \frac{\partial\mathcal{L}_{\text{CL}}}{\partial\boldsymbol{\Theta}_{\text{recog}}}\right\}, \tag{11}$$

where α is a hyperparameter controlling the learning rate.

3. Evaluation and analysis

3.1. Experimental setup

3.1.1. Data preparation

We evaluate our method on two tasks: 1) music-corrupted speech recognition, and 2) two-speaker speech recognition. We used a publicly available corpus AISHELL-1 [32] as the clean speech data set, which contains 400 speakers and 178 hours of Mandarin speech data. The speech data set was split into a training set containing $120,098$ utterances from 340 speakers, a development set containing $14,326$ utterances from 40 speakers, and a test set containing $7,176$ utterances from 20 speakers. We collected an 1100-hour music data set from QQ music broadly covering instrumental music and various genres of English and Mandarin songs. The music data set was first cropped into 10-second music clips, and then split into training, development and test sets following an 8:1:1 ratio.

For the music-corrupted speech experiment, we mixed each speech utterance with a music clip randomly drawn from the corresponding training, development or test split, following signal-to-noise ratio (SNR) from 0 dB to 20dB. Similarly, for the two-speaker experiment, we mixed each speech utterance with an utterance of another speaker randomly drawn from the corresponding split, following SNR from 0dB to 20dB.

3.1.2. Hyperparameter and network architecture

We used 16K sample rate, 25ms frame length, 10ms frame shift, and STFT length of 512 ($F = 257$). For optimization, we set batch size (M) to 24, initial learning rate (α) to 10^{-4} and weight decay to 0.8, and considered the training as converged when the loss on the development set had not improved for 3 consecutive epochs. We set the embedding size (K) to 40, the number of filters (D) to 40, and the single-sided context window size (W) to 5, and the Center Loss weight (λ) to 0.01.

In the *extractor*, we used a BiLSTM of four layers with peephole connection, each with 600 hidden nodes. The last layer of BiLSTM was followed by a full connection layer, mapping 600-dimension hidden vectors to 24000-dimension embedding vectors; In the *adaptor*, we used a BiLSTM of two

Table 1: *WERs (%) on the test set of the two corrupted monaural speech recognition tasks under different SNR conditions*

Approaches	**SNR of music-corrupted speech**					**SNR of two-speaker speech**				
	0dB	**5dB**	**10dB**	**15dB**	**20dB**	**0dB**	**5dB**	**10dB**	**15dB**	**20dB**
1. CL-CLDNN trained on clean speech only	92.9	85.2	72.2	56.3	42.0	99.5	87.7	69.8	51.0	36.7
2. Separately trained DENet and CL-CLDNN	56.3	44.8	36.9	32.0	29.0	86.4	61.4	40.6	32.1	28.4
3. Jointly trained DENet and CL-CLDNN	50.3	40.6	33.3	29.7	25.9	76.6	50.9	27.0	19.5	15.8
4. Jointly trained DENet, fbanks and CL-CLDNN	47.0	38.0	32.0	27.2	25.6	76.3	49.9	26.1	19.3	15.4
5. Our proposed EAR network	**43.0**	**33.8**	**28.5**	**23.3**	**19.9**	**75.4**	**48.2**	**24.6**	**14.1**	**12.0**
6. Our proposed EAR network + speaker tracing	-	-	-	-	-	*61.0*	*43.7*	*21.8*	*13.7*	*11.7*

layers with peephole connection, each with 600 hidden nodes, followed by a mapping to get 40-dimensional fbanks; In the *recognizer*, we used the same CLDNN architecture as in [22]. Batch normalization was applied to the outputs of the convolutional and LSTM layers for faster convergence and better generalization. Our output units were context-dependent phonemes, which has about 12K classes in our Mandarin ASR system.

3.1.3. Baseline system setup

We examined four alternative state-of-the-art ASR architectures as the baseline systems for our experiment. The first system was based on the CL-CLDNN AM trained on clean speech, which achieved 7.825% WER on the clean AISHELL-1 test set. The second system followed the "plug-and-play" strategy [11], i.e., feeding the CL-CLDNN AM (trained on clean speech) with the output of DENet on the corrupted speech, where the extracted spectrogram was converted to fbanks using predefined Mel filtering operations [27]. The third system had DENet and CL-CLDNN jointly trained on the corrupted speech, with no learnable adaptor in between, but it used the predefined differentiable Mel filtering operations instead. The fourth system jointly learned fbanks using a one-layer DNN, as proposed in [11]. Aside from the first system (37.7M parameters), all the systems including our proposed EAR network had comparable model complexity (60.9 $\sim$ 64.9M parameters) that included the speech separation complexity in total.

3.2. Result and discussion

Table 1 shows the word error rates (WERs) on the test set of the music-corrupted speech and the two-speaker speech recognition tasks in five SNR conditions – 0dB, 5dB, 10dB, 15dB, and 20dB. We observe that the WER declines in each column consistently, and in most cases remarkably. This indicates the effectiveness of the elegantly designed solutions for improving joint training. The best results are marked with bold figures, which are achieved consistently by our proposed EAR network, demonstrating its superiority over the four baselines, under all SNR conditions for both tasks. In particular, we are interested in the relative WER reductions by the EAR over the conventional joint training method (the 3rd system), and compare them with the relative WER reductions by the 4th system over the 3rd, as they can reflect the advantage brought by the proposed *adaptor* with more appropriate architecture design and training approach. Figure 2 illustrates these scores, showing that the proposed *adaptor* makes much more significant relative improvements over the conventional joint training approach than the architecture proposed in [11] does, which proves the rationale that we discussed in Section 2.2.

For the two-speaker ASR task, the result shows the EAR has a significant advantage in medium and high SNR condi-

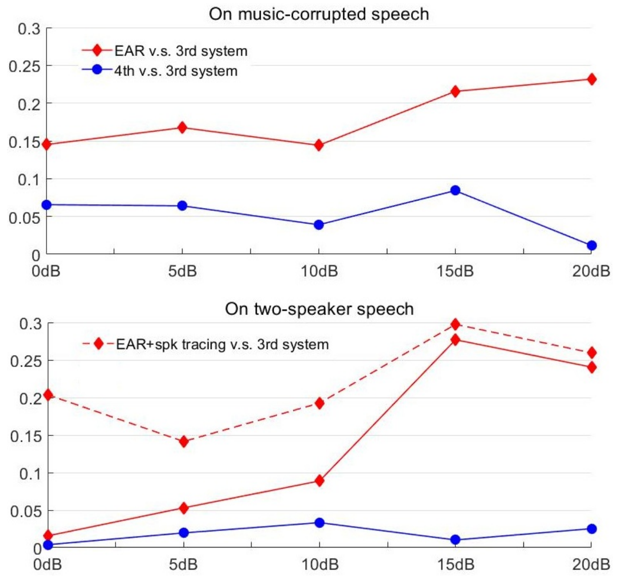

Figure 2: *Relative WER reduction*

tions (10dB to 20dB), however, it is notable that the dominance vanishes in adverse SNR conditions (0dB). This is reasonable as there is no clue for the *extractor* to separate the "target" speaker in the case of 0dB mixtures. To investigate how the EAR performs with speaker tracing, we use an oracle attractor computed with each target speaker's development data set instead of the global average attractor discussed in Section 2.1.1. The resultant WERs are list in the last row in Table 1 and the relative WER reductions are plotted in a dashed red line in Figure 2. Other speaker tracing mechanisms like those adopted in [9, 8, 20] can also be applied here.

4. Conclusions

The contributions of the proposed EAR network can be concluded in two-fold: (1) allowing end-to-end trainable integration of (potentially varied) cutting-edge SS and AM modules to achieve promising WERs in adverse environments, without the expense of compromised performance of any individual module; (2) revealing the importance of an elegant architecture design for bridging the above two modules.

5. Acknowledgements

This research is supported by Hong Kong Research Grants Council General Research Fund No. 14200218 and Shun Hing Institute of Advanced Engineering Project No. MMT-p1-19.

6. References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[3] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[5] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.

[6] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech & Language*, vol. 24, no. 1, pp. 1–15, 2010.

[7] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1670–1679, 2015.

[8] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," *arXiv preprint arXiv:1704.01985*, 2017.

[9] Z. Chen, J. Droppo, J. Li, W. Xiong, Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 1, pp. 184–196, 2018.

[10] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4375–4379.

[11] Z.-Q. Wang and D. Wang, "A joint training framework for robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 796–806, 2016.

[12] J. Du, Q. Wang, T. Gao, Y. Xu, L.-R. Dai, and C.-H. Lee, "Robust speech recognition with speech enhanced deep neural networks," in *INTERSPEECH*, 2014.

[13] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7398–7402.

[14] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 826–835, 2014.

[15] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 297–302.

[16] K. Kenichi, M. Wu, S. Shiva, S. Nikko, and H. Bjrn, "Multi-geometry spatial acoustic modeling for distant speech recognition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[17] M. Wu, K. Kenichi, S. Shiva, S. Nikko, and B. Hoffmeister, "Frequency domain multi-channel acoustic modeling for distant speech recognition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[18] S. Paris, F. Cedric, J. M. Gautham, M. Nasser, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Magazine*, vol. 31, pp. 66–75, 2014.

[19] J. M. Gautham and S. Paris, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011.

[20] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," *arXiv preprint arXiv:1807.08974*, 2018.

[21] J. Wang, D. Su, J. Chen, S. Feng, D. Ma, N. Li, and D. Yu, "Learning discriminative features in sequence training without requiring framewise labelled data," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[22] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[23] 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.

[24] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[26] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[27] J. Lyons, "python_speech_features," https://github.com/jameslyons/python_speech_features, 2013.

[28] G. Alex, W. Greg, and I. Danihelka, "Neural turing machines," https://arxiv.org/abs/1410.5401, 2014.

[29] H. Liu, W. Shi, W. Huang, and Q. Guan, "A discriminatively learned feature embedding based on multi-loss fusion for person search," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1668–1672.

[30] H. M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, *Automatic differentiation: applications, theory, and implementations*. Springer Science & Business Media, 2006, vol. 50.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[32] X. N. B. W. H. Z. Hui Bu, Jiayu Du, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA 2017*, 2017, p. Submitted.