JOINT NOISE ADAPTIVE TRAINING FOR ROBUST AUTOMATIC SPEECH RECOGNITION

Arun Narayanan[⋆] *and DeLiang Wang*[⋆†]

[⋆]Department of Computer Science and Engineering
[†]Center for Cognitive and Brain Sciences
The Ohio State University
Columbus, OH 43210-1277, USA
{narayaar,dwang}@cse.ohio-state.edu

ABSTRACT

We explore time-frequency masking to improve noise robust automatic speech recognition. Apart from its use as a frontend, we use it for providing smooth estimates of speech and noise which are then passed as additional features to a deep neural network (DNN) based acoustic model. Such a system improves performance on the Aurora-4 dataset by 10.5% (relative) compared to the previous best published results. By formulating separation as a supervised mask estimation problem, we develop a unified DNN framework that jointly improves separation and acoustic modeling. Our final system outperforms the previous best system on CHiME-2 corpus by 22.1% (relative).

Index Terms— Deep neural network, noise robustness, time-frequency masking, CHiME-2, Aurora-4

1. INTRODUCTION

Automatic speech recognition (ASR) has come a long way in the last few years, especially after the introduction of deep neural network based acoustic models (DNN-AMs) [1]. With systems achieving acceptable performances in relatively *clean* conditions, focus on robustness has been sharpened. In this work, we mainly consider noise robustness, which has been widely studied over the last two decades [2]. When computational complexity is not a main concern, model adaptation techniques perform well (e.g., [3]). But most such techniques assume Gaussian mixture model (GMM) based acoustic models (GMM-AMs); it is currently unclear how to extend them to DNN based systems. Feature adaptation techniques (e.g., [4, 5]), on the other hand, do not make any assumptions about the ASR backend and are, therefore, more directly applicable.

A class of feature adaptation techniques is based on time-frequency (T-F) masking for speech separation. In such methods, a T-F mask is estimated from the input signal which is then used to enhance the noisy spectrogram (or any other T-F representation that is used) [6]. ASR features are subsequently extracted from the enhanced spectrogram. Recently, data-driven mask estimation algorithms have shown a lot of promise [7]. Our investigation of such techniques [8, 9] showed that they work reasonably well even when DNN based acoustic models are used. But the improvement compared to a GMM based backend is significantly lower. In this work we further explore such methods. Specifically, we focus on two aspects. Firstly, we study if there are alternative ways of using the output of speech separation to improve ASR performance. Secondly, since the separation system is supervised, we study training strategies that unify separation and the backend acoustic modeling.

This paper is organized as follows. Section 2 describes prior work. A preliminary study on using the output of separation in novel ways to improve ASR is described in Section 3. Our system, which unifies separation and acoustic modeling, is presented in Section 4, followed by results in Section 5. Section 6 concludes the paper.

2. RELATION TO PRIOR WORK

The proposed system uses DNN based acoustic models which have been shown to work well in the presence of noise as long as the mismatch between training and testing conditions is not significant [10, 8, 11]. With a mismatch, feature enhancement/separation has been shown to be useful [8, 11]. The current work builds upon our previous work in [8], which uses an ideal ratio mask (IRM) based frontend defined for a mel-spectral representation of speech. In [8], the IRM is estimated by combining subband and fullband DNNs using a set of features extracted specifically for the purpose. As shown in Section 4, the current work significantly simplifies that system. Moreover, in [8], IRM estimation and acoustic modeling are done independent of one another, unlike the proposed system.

Noise-aware training (NAT) was proposed in [10] to improve noise robustness of DNN based ASR systems. In addition to the noisy log compressed mel-spectrogram (log-MS), it uses a crude estimate of noise obtained by averaging the first and the last few frames of each utterance as input to the DNN-AMs. This improved performance on the Aurora-4 noisy ASR task [12] by 3.9% (relative). Our system uses a similar strategy. But, instead of using a crude estimate, we use speech separation to obtain a more accurate estimate of noise. We also use additional features derived from speech separation to further improve performance.

With DNNs, it has been observed that using a speech separation frontend does not always improve performance [10, 8], especially when log mel-spectral features, which have been shown to work better than cepstral features [13, 8], are used as input. A strategy commonly used with GMMs is retraining the ASR system using the enhanced features to reduce mismatch [14]. But frontends invariably introduce distortions, and with DNN-AMs retraining can sometimes negatively affect performance [8]. An alternative strategy with GMM-AMs has been joint or adaptive training – the enhancement and recognition modules are optimized jointly [15, 16, 17]. A probabilistic formulation of both the enhancement frontend and the GMM-AMs lends itself to EM-style iterative training. In the context of noise robustness, such joint training strategies, to the best

This research was supported in part by an AFOSR grant (FA9550-12-1-0130), an NIDCD grant (R01 DC012048), and the Ohio Supercomputer Center.

of our knowledge, have not been proposed for DNN-AMs. In this study, we propose a novel joint training algorithm for DNN-AMs; we call it joint noise adaptive training for DNN-AMs (D-JNAT).

3. ALTERNATIVE FEATURES FROM SEPARATION

We will first look at alternative ways of using the output of speech separation to improve ASR performance. Given an estimate of the ideal ratio mask (IRM), which is defined as the ratio of speech to mixture energy at each T-F unit assuming uncorrelated noise, we can obtain an estimate of both speech and noise as follows:

$$\widehat{\mathbf{N}}(t) = (1 - \widehat{\mathbf{M}}(t)) \odot \mathbf{Y}(t), \quad (1)$$

$$\widetilde{\mathbf{X}}(t) = (\widehat{\mathbf{M}}(t))^{\alpha} \odot \mathbf{Y}(t), \quad (2)$$

$$\widehat{\mathbf{X}}(t) = f(\widetilde{\mathbf{X}}(t), \mathbf{Y}(t)). \quad (3)$$

Here, $\mathbf{Y}$ is the mixture mel-spectrogram (MS), $\widehat{\mathbf{X}}$, $\widetilde{\mathbf{X}}$, and $\widehat{\mathbf{N}}$ correspond to estimates of the clean, noise-removed, and noise MS. $\widehat{\mathbf{M}}$ is an estimate of the IRM, and has values in the range $[0, 1]$. t indexes time-frames. α is a tunable parameter (< 1) that exponentially scales-up IRM estimates, thereby reducing the distortion introduced by masking. Exponentiation is done point-wise. In these preliminary experiments, α is set to 1. $\odot$ denotes point-wise multiplication. $f(\cdot)$ is a function that undoes the distortion introduced by channel or microphone mismatch between training and testing. It may take additional features as input, apart from those shown in Eq. 3. For example, in [8], $f(\cdot)$ is learned in a supervised fashion using DNNs that take a window of the noisy and noise-removed MS as inputs. Given these estimates, we look at the following alternative feature representations to train a DNN-AM:

- Noisy mel-spectrogram (NMS): 26-channel noisy log-MS along with the first and second derivatives. The features are derived by splicing together the log-MS for 11 contiguous time-frames after sentence level mean normalization. This is the standard feature set used by most DNN systems, and forms our baseline.
- Noisy mel-spectrogram + noise estimate (1) (NMS + NE(1)): This feature set replicates the system proposed in [10]. The noise estimate at each time frame is obtained by averaging the first and the last 15 frames of the noisy log-MS.
- Noisy mel-spectrogram + noise estimate (2) (NMS + NE(2)): Same as above, but the noise estimate is obtained from Eq. 1 after smoothing it using a 9th order ARMA filter [18].
- Noisy mel-spectrogram + noise estimate (2) + speech estimate (NMS + NE(2) + SE): Same as above, but additionally uses a speech estimate which is obtained by smoothing $\widehat{\mathbf{X}}$ in Eq. 3 using a 2nd order ARMA filter.
- Noisy mel-spectrogram + noise estimate (2) + residual noise estimate (NMS + NE(2) + RNE): A crude residual noise estimate, which may also carry some channel information and is obtained by averaging the first and last 15 frames of $\widetilde{\mathbf{X}}$ in Eq. 2, is additionally used as a feature.

The results obtained on the Aurora-4 dataset using these alternative features are shown in Table 1. A DNN-AM with 7 hidden layers and trained with dropout [19] is used (see Section 4.2 for details). The ratio mask estimate and the feature mapping function $f(\cdot)$ are the same as in [8]. Several interesting observations come out of this preliminary study. Firstly, our results are better than those reported in [10]. This is partly because of the training recipe that we used and also because our training labels are obtained by aligning the clean training set. With this improvement, NAT ('NMS + NE(1)' in table) proposed in [10] does not seem to have a lot of effect in performance. Only in clean+channel mismatched condition does it improve performance slightly. Note that most of the noises in Aurora-4 can be categorized as non-stationary, contrary to the assumption made by NAT. Using a more accurate noise estimate that comes from Eq. 1 improves performance in noisy+channel mismatched conditions by 0.4 percent. The average performance improves by 0.2 percent. Adding the smoothed speech estimate improves performance in noisy conditions by 0.4 percent, but lowers performance in noisy+channel mismatched conditions by 1.0 percent. This is similar to the trend we noticed when using only the speech estimate as an additional feature in [8]; the distortions introduced by separation seems to have a detrimental effect in the presence of noise+channel mismatch. Adding the residual noise estimate improves performance in noisy+channel mismatched conditions by 0.1 percent, but deteriorates performance in noisy conditions by 0.2 percent. As was suggested in [20], we tried averaging the posteriors obtained from 'NMS + NE(2) + SE' and 'NMS + NE(2) + RNE' systems as they perform the best in noisy and noisy+channel mismatched conditions, respectively (POS_AVG). Interestingly, POS_AVG retains the best performance in both these conditions and improves upon the previous best results on this corpus by 10.5% [10], and 8.3% (relative) [8][1].

Table 1. Word error rates (WER) on the Aurora4 corpus using alternative features. NMS, which forms our baseline, stands for noisy log-mel-spectrogram. NE, SE, RNE, and POS_AVG stand for noise, speech, and residual noise estimate, and posterior averaging, respectively (see text for details). The columns Clean, Noisy, Clean + Channel, and Noisy + Channel correspond to the WER averaged on test sets 1, 2 to 7, 8, and 9 to 14, respectively, of Aurora-4.

System	Clean	Noisy	Clean + Channel	Noisy + Channel	Average
NMS	4.8	7.9	7.9	17.4	11.7
+ NE(1)	4.7	7.9	**7.6**	17.4	11.7
+ NE(2)	4.6	7.8	7.9	17.0	**11.5**
+ SE	5.0	**7.4**	8.4	18.0	11.8
+ RNE	**4.5**	8.0	8.1	16.9	11.5
POS_AVG	**4.5**	**7.4**	8.1	**16.5**	**11.1**

4. SYSTEM DESCRIPTION

Arguably, channel mismatch is easier to handle as we can collect more data to cover additional devices (e.g., cell-phones) on which the ASR system needs to be deployed. In comparison, noise is much more unpredictable. For the D-JNAT system that we present here, it is assumed that there is not a lot of channel mismatch between training and testing, and that speech separation primarily addresses noise ($f(\cdot)$ in Eq. 3 deterministically maps $\widetilde{\mathbf{X}}(t)$ to $\widehat{\mathbf{X}}(t)$). A block diagram of the proposed system is shown in Figure 1. The components of the system are described in detail below.

4.1. Speech separation

As mentioned, speech separation is done via ratio masking in the mel-spectral domain. We use a 26-channel MS that spans frequencies in the range 50 Hz to 7 kHz. A window size of 20 msec and a hop size of 10 msec are used. The IRM is estimated using a system similar to [8], but is simplified so that it can be easily incorporated

[1] [8] performs online feature adaptation, whereas [10] does not.

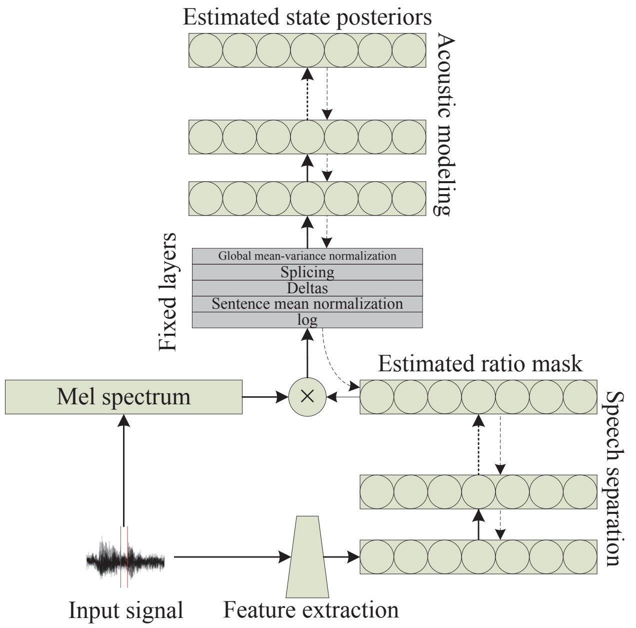

Fig. 1. A block diagram of joint adaptive training. The two main components of the system, speech separation and acoustic modeling, are shown along with how they are joined into a single framework.

into the joint framework. The following feature set is extracted at every time frame from the noisy input signal:

- 13 dimensional RASTA filtered perceptual linear predictive cepstral coefficients (RASTA-PLPs) [21]. The features for 7 contiguous frames are spliced together to add context.
- Amplitude modulation spectrograms (AMS) [22]. 15-dimensional AMS features are extracted separately for each of the 26 frequency bands in the MS. They are then concatenated to form the input feature for a time-frame.
- 31 dimensional broadband and narrowband mel frequency cepstral coefficients (MFCCs). Narrowband MFCCs are extracted using an analysis window of 200 msec [23] and add a lot more context than braodband MFCCs, which use a 20 msec window. Similar to RASTA-PLPs, the MFCC features of 7 contiguous frames are spliced together to form the input representation.

The above features are concatenated together to form a 915-dimensional (13×7 + 15×26 + 31×2×7) input feature, which is then fed to a 3 hidden layer DNN that estimates the IRM for all 26 frequency channels. Each hidden layer has 1024 nodes. The DNN is trained with a dropout rate of 0.3. The hidden nodes use rectified linear activations (ReLU) and the output nodes sigmoidal activations. The weights are learned using mini-batch stochastic gradient descent with adagrad and momentum. The momentum is linearly increased from 0.1 to 0.5 over the first 5 epochs after which it is set to 0.9. Mini-batch size is set to 256. The weights are initialized at random; no RBM-pretraining is used. We also normalize the L_2 norm of incoming weights of each hidden node to 1 [19]. The DNN is trained for 50 epochs to minimize the cross-entropy error criterion. The learning rate is set to 0.01 for the first 10 epochs, 0.005 for the next 20 epochs, and 0.001 for the last 20 epochs.

Note that [8] estimates masks at the subband and the fullband levels, and then combines these estimates over a window to explicitly incorporate context. The proposed system, on the other hand, incorporates context at the feature-level. Further, the proposed system directly estimates the IRM instead of a transformed version of it as done in [8], which significantly simplifies joint training.

4.2. Acoustic modeling

To train the DNN-AMs, two input feature representations are chosen from those presented in Section 3. The first one corresponds to the 'NMS' feature. The second feature representation corresponds to the 'NMS + NE(2) + SE' feature as it performed the best in the presence of noise (cf. Table 1). The noise and speech estimates are obtained in a similar fashion using the IRM estimated by the system described in Section 4.1. The DNN-AMs consist of 7 hidden layers, each with 2048 nodes, and are trained similar to the DNNs used for IRM estimation. The output layer uses softmax activation. The learning rate is set to 0.005 for the first 30 epochs and 0.001 for the final 20 epochs; the rest of the parameters remain unchanged.

4.3. Joint training

The main goal behind joint training is to unify separation and acoustic modeling. Typically, the output of separation undergoes further processing before it is fed to the acoustic model. In our joint system, we model these processing steps as *fixed* hidden layers of a single deep network. They are shown in gray in Figure 1 and includes operations like log-compression, feature normalization, delta calculation, and feature splicing. Interestingly, all of these operations can be performed within a DNN framework using appropriate weights and/or network architectures. For example, delta features can be calculated using a linear activation layer with weights as below [24]:

$$\begin{bmatrix} \mathbf{o}_t \\ \triangle \mathbf{o}_t \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \mathbf{I} & 0 \\ -\mathbf{I} & 0 & \mathbf{I} \end{bmatrix}}_{\mathbf{W}_\triangle} \begin{bmatrix} \mathbf{o}_{t-1} \\ \mathbf{o}_t \\ \mathbf{o}_{t+1} \end{bmatrix} \quad (4)$$

Here, $\mathbf{o}_t$ is the static feature at time t, $\mathbf{I}$ is the identity matrix, and $\mathbf{W}_\triangle$ is the desired weight matrix of the hidden layer. The above formulation calculates deltas over a window of 3 frames, but it can be extended trivially to more than 3 frames and for calculating double-deltas. Further, the connections from the preceding layer can be modified so that this layer receives static features from multiple, contiguous time frames as is necessary for delta calculation. The other fixed layers can be modeled in a similar fashion.

With the above formulation, it is easy to see that the two modules can now be trained jointly; with the fixed hidden layers cast as a 'neural network', the error gradients from the DNN-AM can flow through them back to the separation module. While training such a system, we initialize both the acoustic model and the IRM estimator using the independently trained DNNs. The trainable weights of both networks are then tuned together for a few additional epochs.

We apply joint training to the DNNs described in the previous subsections. The following systems are considered:

- DNN-AM that uses the 'NMS' feature as its input. The IRM estimated by the separation module enhances the NMS features using Eq. 2 which is then used as input to the DNN-AM. Joint training is used to further enhance the NMS feature and adapt the DNN-AM. This will be referred to as joint adaptive training (D-JAT).
- DNN-AM that uses the 'NMS + NE(2) + SE' feature as its input. The noise and speech estimates are obtained using the initial IRM estimator which is trained independent of the DNN-AM. Joint training is used to enhance the NMS feature and the DNN-AM (D-JNAT).

Table 2. WER on the CHiME-2 corpus. NMS, which forms our baseline, stands for noisy log-mel-spectrogram. ERM, NE, SE, D-JAT, and D-JNAT stand for estimated ratio mask, noise estimate, speech estimate, joint adaptive training, and joint noise adaptive training, respectively (see text for details). The previous best results on this corpus is also shown.

System	**si_dt_05**	**si_et_05**						
	Average	**-6 dB**	**-3 dB**	**0 dB**	**3 dB**	**6 dB**	**9 dB**	**Average**
NMS	28.9	37.9	30.1	25.9	21.1	18.3	16.5	25.0
ERM	26.6	33.0	26.6	23.9	19.5	16.6	15.8	22.6
+ NE(2) + SE	25.9	33.2	26.7	22.4	18.7	15.4	14.5	21.8
D-JAT	25.8	32.1	25.6	23.1	18.6	16.1	15.1	21.7
D-JNAT	25.1	31.4	24.8	21.4	18.1	15.1	14.1	20.8
[14]	32.9	42.7	33.9	27.5	21.8	18.4	16.2	26.7

5. RESULTS

5.1. Experimental setup

The proposed system is evaluated on the CHiME-2 corpus [25]. It is a medium-large vocabulary task based on the *Wall Street Journal* corpus (WSJ0). The training and the test conditions simulate a family living room. The utterances are reverberant, and are mixed at signal-to-noise ratios (SNR) in the range $[-6, 9]$ dB. Even though the corpus is binaural, our system is currently monaural; we simply average the left and right ear recordings for all experiments. The IRM estimator needs noisy and the corresponding clean recordings to define the targets at the time of training. Since such recordings are not provided with the corpus, we artificially mix the reverberant noise-free utterances in the training set with randomly selected segments of the noise recordings provided with the corpus. The fraction of recordings at each SNR is the same as in the official noisy-reverberant training set. This new set is used *only* to train the IRM estimator when it is trained independent of the DNN-AM.

In order to obtain senone (or tied-triphone state) labels for training the DNN-AM, an ML trained GMM-HMM system is used. The clean training set of WSJ0 is used to train and subsequently align the utterances. Based on the pruning parameters, the system ended up with 3298 senones. The DNN-AMs are trained using the official noisy-reverberant training set. The joint systems (D-JAT and D-JNAT) are initialized using the independently trained IRM estimator and DNN-AM, and jointly trained for 10 epochs. A mini-batch size of 512 and a learning rate of 0.001 is used. For D-JAT and D-JNAT, the final model is chosen based on the WER on the official development set (si_dt_05). The development set is also used to choose α in Eq. 2, the chosen value being 0.5. A subset of this development set is used to choose some of the hyper-parameters while training, like the learning rates and the value at which the log-gradient is clipped to prevent it from dominating the error gradient. The feature normalization parameters are re-calculated after every epoch.

5.2. Evaluation results

For limitations of space, we only present detailed results on the final test set (si_et_05) along with the average results on the development set. The results are shown in Table 2. As can be seen, our baseline system trained using noisy log-MS features gives an average WER of 25.0 percent, which is in itself better than the previous best results on this corpus by 6.7% (relative) [14]. The system in [14] uses bidirectional long short-term memory based feature enhancement and a discriminatively trained, speaker adapted GMM-HMM system. Interestingly, an unadapted DNN-HMM system is able to outperform such a system. Using the estimated ratio mask (ERM) to enhance the noisy speech (see Eq. 2) improves performance by 2.4 percent (absolute) compared to the baseline. Note that the ASR models are not retrained using the masked speech; doing so did not improve performance (results not in the table). D-JAT, which jointly trains the IRM estimator and the DNN-AM, improves performance by another 0.9 percent compared to ERM. The 'NMS + NE(2) + SE' system improves performance on the development set by 1.7 percent (results not in the table) compared to the NMS baseline. When the ERM is used to enhance the noisy log-MS used by the system, WER reduces by another 1.3 percent. On the test set, such a system performed similar to D-JAT, as shown in the table. Our final system, D-JNAT, produces an average WER of 20.8 percent, 1 percent (absolute) better than the system that does not use joint training, and 4.2 percent better than our baseline (6.5 percent at -6 dB). This is also a 22.1% (relative) improvement over the system in [14]. It is worth mentioning that NAT [10] did not improve performance on this corpus compared to the NMS baseline; this is to be expected as noise used by the corpus is highly non-stationary.

It was observed that the masks generated by the jointly trained model attenuate noise a lot more than those generated by the independently trained models, while preserving spectro-temporal patterns that are most important for recognition. There is a disconnect between the criterion that is commonly used for mask estimation (SNR improvement) and ASR (WER reduction), and our past work has shown that SNR improvements and ASR performance (or speech intelligibility) are not fully correlated [26]. Joint training, on the other hand, directly optimizes a criterion that is important to improve ASR. Since ASR and speech intelligibility tend to correlate [26], such joint training schemes may provide an alternative, more useful criterion to optimize for speech separation algorithms that focus on intelligibility. Note that the proposed training strategy is quite flexible, and can potentially by used by other separation frontends like feature mapping [5, 14, 8].

6. CONCLUSION

We have proposed novel ways for improving the state-of-the-art in noise robust ASR using time-frequency masking. By using speech separation to provide smooth estimates of speech and noise to a DNN-AM, we are able to significantly improve performance in a wide range of conditions. Modeling separation and recognition in a unified framework yields a novel joint noise adaptive training strategy to optimize the parameters of both systems, which further improves performance. The results obtained using our system represent the best published in two commonly used medium-large vocabulary ASR tasks – Aurora-4 and CHiME-2. Going forward, we plan to incorporate discriminative training [27] into our framework to allow higher-level sequence structure to influence separation and acoustic modeling.

7. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. W. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] T. Virtanen, B. Raj, and R. Singh, Eds., *Techniques for Noise Robustness in Automatic Speech Recognition*, John Wiley & Sons, West Sussex, UK, 2012.

[3] Y.-Q. Wang and M. J. F. Gales, "Speaker and noise factorization for robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2149–2158, 2012.

[4] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4101–4104.

[5] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng., "Recurrent neural networks for noise reduction in robust ASR," in *Proceedings of Interspeech*, 2012.

[6] D. L. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Wiley/IEEE Press, Hoboken, NJ, 2006.

[7] Y. Wang and D. L. Wang, "Feature denoising for speech separation in unknown noisy environments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7472–7476.

[8] A. Narayanan and D. L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," Tech. Rep. OSU-CISRC-6/13-TR14, Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA, 2013, Available: ftp://ftp.cse.ohio-state.edu/pub/tech-report/2013/TR14.pdf.

[9] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7092–7096.

[10] M. L. Seltzer, D. Yu, and Y.-Q. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7398–7402.

[11] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *Proceedings of Interspeech*, 2013, pp. 2992–2996.

[12] N. Parihar and J. Picone, "Analysis of the Aurora large vocabulary evalutions," in *Proceedings of the European Conference on Speech Communication and Technology*, 2003, pp. 337–340.

[13] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14 –22, 2012.

[14] F. Weninger, J. Geiger, M. Wllmer, B. Schuller, and G. Rigoll, "The Munich feature enhancement approach to the 2nd CHiME challenge using BLSTM recurrent neural networks," in *Proceedings of the 2nd CHiME workshop on machine listening in multisource environments*, 2013, pp. 86–90.

[15] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proceedings of the Fourth International Conference on Spoken Language*, 1996, vol. 2, pp. 1137–1140.

[16] H. Liao and M. J. F. Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, vol. 4, pp. 389–392.

[17] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3825–3828.

[18] C.-P. Chen and J. A. Blimes, "MVA processing of speech features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 257–270, 2007.

[19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[20] B. Li and K. C. Sim, "Improving robustness of deep neural networks via spectral masking for automatic speech recognition," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 279–284.

[21] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578 –589, 1994.

[22] B. Kollmeier and R. Koch, "Speech enhancement based on physiological and psychoacoustical models of modulation perception and binaural interaction," *Journal of Acoustical Society of America*, vol. 95, pp. 1593–1602, 1994.

[23] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," under review.

[24] R. C. Van Dalen and M. J. F. Gales, "Extended VTS for noise-robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 733–743, 2011.

[25] E. Vincent, J. Barker, S. Watanabe, J. LeRoux, F. Nesta, and M. Matassoni, "The 2nd chime speech separation and recognition challenge," 2012.

[26] A. Narayanan and D. L. Wang, "The role of binary mask patterns in automatic speech recogniton in background noise," *Journal of Acoustical Society of America*, vol. 133, pp. 3083–3093, 2013.

[27] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3761–3764.