Robust Speech Recognition with Speech Enhanced Deep Neural Networks

Jun Du[1], *Qing Wang*[1], *Tian Gao*[1], *Yong Xu*[1], *Lirong Dai*[1], *Chin-Hui Lee*[2]

[1]University of Science and Technology of China, Hefei, Anhui, P.R. China
[2]Georgia Institute of Technology, Atlanta, GA. 30332-0250, USA

{jundu,lrdai}@ustc.edu.cn, {xiaosong,gtian09,xuyong62}@mail.ustc.edu.cn, chl@ece.gatech.edu

Abstract

We propose a signal pre-processing front-end to enhance speech based on deep neural networks (DNNs) and use the enhanced speech features directly to train hidden Markov models (HMMs) for robust speech recognition. As a comprehensive study, we examine its effectiveness for different acoustic features, acoustic models, and training-testing combinations. Tested on the Aurora4 task the experimental results indicate that our proposed framework consistently outperform the state-of-the-art speech recognition systems in all evaluation conditions. To our best knowledge, this is the first showcase on the Aurora4 task yielding performance gains by using only an enhancement pre-processor without any adaptation or compensation post-processing on top of the best DNN-HMM system. The word error rate reduction from the baseline system is up to 50% for clean-condition training and 15% for multi-condition training. We believe the system performance could be improved further by incorporating post-processing techniques to work coherently with the proposed enhancement pre-processing scheme.

Index Terms: robust speech recognition, speech enhancement, clean-condition training, multi-condition training, hidden Markov models, deep neural networks

1. Introduction

With the fast development of mobile internet, the speech-enabled applications using automatic speech recognition (ASR) are becoming increasingly popular. However, the noise robustness is one of the critical issues to make ASR system widely used in real world. Historically, most of ASR systems use Mel-frequency cepstral coefficients (MFCCs) and their derivatives as speech features, and a set of Gaussian mixture continuous density HMMs (CDHMMs) for modeling basic speech units. Many techniques [1, 2, 3] have been proposed to handle the difficult problem of mismatch between training and application conditions. One type of approaches to dealing with the above problem is the so-called data-driven approach based on stereo-data, which is also the topic of this study. SPLICE [4] is one successful showcase which is a feature compensation approach by using environmental selection and stereo data to learn the mapping function between clean speech and noisy speech via Gaussian mixture models (GMMs). Then similar approaches are proposed in [5, 6]. In [7], a stereo-based stochastic mapping (SSM) technique is presented, which outperforms SPLICE. The basic idea of SSM is to build a GMM for the joint distribution of the clean and noisy speech by using stereo data. To relax the constraint of recorded stereo-data, we propose to use synthesized pseudo-clean features generated by exploiting HMM-based synthesis to replace the ideal clean features from one of the stereo channels in SPLICE and SSM [8, 9].

The recent breakthrough of deep learning [10, 11], especially the application of deep neural networks (DNNs) in ASR area [12, 13, 14], marks a new milestone that DNN-HMM for acoustic modeling becomes the-state-of-the-art instead of GMM-HMM. It's believed that the first several layers of DNN play the role of extracting highly nonlinear and discriminative features which are robust to irrelevant variabilities. This makes DNN-HMM inherently noise robust to some extent, which is verified on Aurora4 database in [15]. In [16, 17], several conventional front-end techniques can further yield performance gain on top of DNN-HMM system for tasks with small vocabulary or constrained grammar. But on large vocabulary tasks, the traditional enhancement approach as in [18] which is effective for GMM-HMM system may even lead to the performance degradation for DNN-HMM system with log Mel-filterbank (LMFB) features under the well-matched training-testing condition [15]. Meanwhile, the data-driven approaches using stereo-data via recurrent neural network (RNN) and DNN proposed in [19, 20] can improve the recognition accuracy on small vocabulary tasks. More recently, the masking techniques [21, 22, 23] are successfully applied for noisy speech recognition. In [23], the approach using time-frequency masking combined with feature mapping via DNN and stereo-data claims to achieve the best results on Aurora4 database. Unfortunately, for multi-condition training using DNN-HMM with LMFB features, this approach still results in worse performance, which is similar to the conclusion in [15].

In this study, inspired by our recent progress on speech enhancement via DNN as a regression model [24], we further verify its effectiveness for noisy speech recognition. First, DNN is adopted as a pre-processor, which directly estimates the complicated nonlinear mapping from observed noisy speech with acoustic context to desired clean speech in log-power spectral domain. Second, we propose to use global variance equalization (GVE) to alleviate the over-smoothing problem of DNN based regression model, which is implemented as a post-processing operation by linear scaling of log-power spectral features. Third, an exhaustive experimental study is conducted by the comparison of different acoustic features (MFCC and LMFB), acoustic models (GMM-HMM and DNN-HMM), and training-testing conditions (high-mismatch, mid-mismatch, and well-matched). Our approach achieves promising results on Aurora4 database for all testing cases. Furthermore, compared with the enhancement approaches in [15, 23], this is the first time to yield performance gain by using our proposed approach for the multi-condition training with LMFB features and DNN-HMM on Aurora4 database, which indicates that the proposed front-end DNN can further improve the noise robustness on top of DNN-HMM systems under the well-matched condition for large vocabulary tasks.

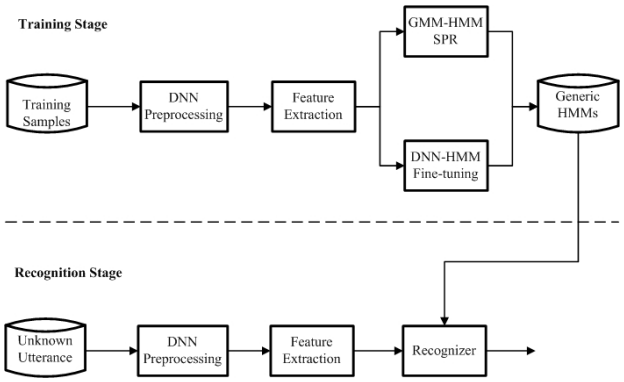

Figure 1: Overall development flow and architecture.

2. System Overview

The overall flowchart of our proposed ASR system is illustrated in Fig. 1. In the training stage, first the training samples are pre-processed by DNN based speech enhancement in the log-power spectral domain. Then enhanced spectra are further processed to extract the acoustic features, namely LMFB or MFCC features with cepstral mean normalization (CMN), which are adopted to train the generic HMMs. For GMM-HMM system, single pass retraining (SPR) [28] is used to generated the generic models. The SPR works as follows: given one set of well-trained models, a new set matching a different training feature type can be generated in a single re-estimation pass, which is done by computing the forward and backward probabilities using the original models together with the original training features and then switching to the new training features to compute the parameter estimation for the new set of models. In our case, the original model and training features are generated using clean-condition training data of Aurora4 database while the new features refer to enhanced features. Obviously, SPR is a simpler and faster training procedure than the traditional retraining of GMM-HMMs using the new features from scratch. Our experiments also confirm that SPR can achieve better recognition performance.

As for DNN-HMM system, we design a novel procedure for the training of DNN acoustic model with enhanced features. Prior to this, a reference DNN should be trained using original features without DNN pre-processing via the procedure in [12]. First, with the well-trained GMM-HMMs using clean-condition training features, state-level forced-alignment performed to obtain the frame-level labels which is used for DNN training with all kinds of input features, including clean-condition training features, multi-condition training features, and enhanced training features. The training of reference DNN consists of unsupervised pre-training and supervised fine-tuning. The pre-training treats each consecutive pair of layers as a restricted Boltzmann machine (RBM) while the parameters of RBM are trained layer by layer with the approximate contrastive divergence algorithm [11]. After pre-training for initializing the weights of the first several layers, a supervised fine-tuning of the parameters in the whole neural network with the final output layer is performed via the frame-level cross-entropy criterion. On top of this reference DNN as an initialization, the DNN model of enhanced features can be further optimized by only changing the input of DNN from original features to enhanced features. This simple fine-tuning procedure of DNN is not only faster than re-training from scratch but also generates

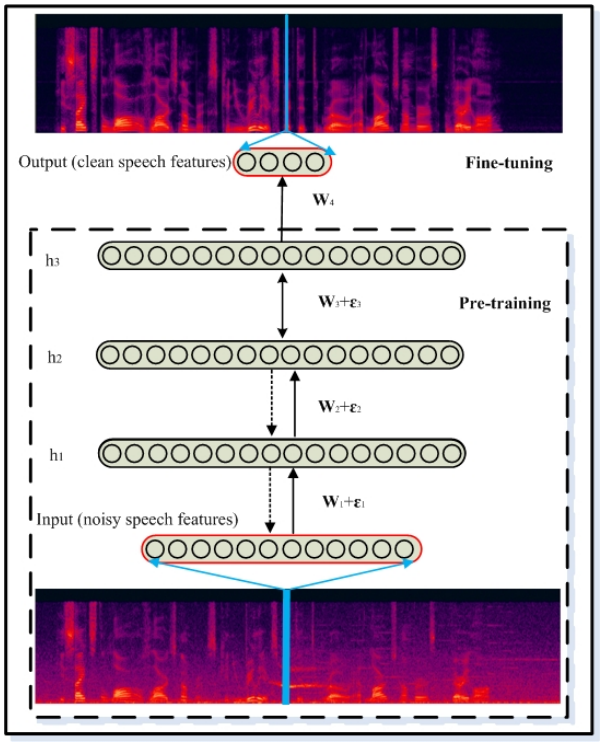

Figure 2: DNN for speech enhancement.

better recognition performance which we explain as the information of original features may have some complementary effects to the imperfectly enhanced feature which can be utilized by powerful DNN modeling.

In the recognition stage, after DNN pre-processing and feature extraction of the unknown utterance, the normal recognition is conducted. In the next section, the details of DNN pre-processor are elaborated.

3. DNN as a Pre-processor

As a pre-processor, DNN is adopted as a regression model, rather than the classification model used in acoustic modeling, to predict the clean log-power spectral features given the input noisy log-power spectral features with acoustic context, which is shown in Fig. 2. The reason why we use log-power spectral features rather than LMFB or MFCC features is all the speech information can be retained in this domain and good listening quality can be obtained from the reconstructed clean speech according to [24]. The acoustic context information along both time axis (with multiple neighboring frames) and frequency axis (with full frequency bins) can be fully utilized by DNN to improve the continuity of estimated clean speech. As the training of this regression DNN requires a large amount of time-synchronized stereo-data with clean and noisy speech pairs, which are difficult and expensive to be collected from real scenarios, the noisy speech utterances are synthesized by corrupting the clean speech utterances with additive noises with different types and SNRs or convolutional (channel) distortions. The training of regression also consists of unsupervised pre-training and supervised fine-tuning. The pre-training is the same as that in DNN for acoustic modeling. For the supervised fine-tuning, we aim at minimizing mean squared error between the DNN

output and the reference clean features:

$$E = \frac{1}{N}\sum_{n=1}^{N} \|\hat{\boldsymbol{x}}_n(\boldsymbol{y}_{n-\tau}^{n+\tau}, \boldsymbol{W}, \boldsymbol{b}) - \boldsymbol{x}_n\|_2^2 + \kappa\|\boldsymbol{W}\|_2^2 \quad (1)$$

where $\hat{\boldsymbol{x}}_n$ and $\boldsymbol{x}_n$ are the n^{th} D-dimensional vectors of estimated and reference clean features, respectively. $\boldsymbol{y}_{n-\tau}^{n+\tau}$ is a $D(2\tau+1)$-dimensional vector of input noisy features with neighbouring left and right τ frames as the acoustic context. $\boldsymbol{W}$ and $\boldsymbol{b}$ denote all the weight and bias parameters. κ is the regularization weighting coefficient to avoid over-fitting. The objective function is optimized using back-propagation procedure with a stochastic gradient descent method in mini-batch mode of N sample frames. Based on our preliminary experiment, we observe that the estimated clean speech has a muffling effect when compared with reference clean speech. To alleviate this problem, GVE, as a post-processing, is used to further enhance the speech region and suppress the residue noise of the recovered speech simultaneously. In GVE, a dimension-independent global equalization factor β can be defined as:

$$\beta = \sqrt{\frac{GV_{\text{ref}}}{GV_{\text{est}}}} \quad (2)$$

where GV_{ref} and GV_{est} are the dimension-independent global variance of the reference clean features and the estimated clean features, respectively. Then the post-processing is:

$$\hat{\boldsymbol{x}}_n^{'} = \beta\hat{\boldsymbol{x}}_n \quad (3)$$

where $\hat{\boldsymbol{x}}_n^{'}$ is the final estimated clean speech feature vector. This simple operation is verified to improve the overall listening quality.

4. Experiments

4.1. Experimental Setup

Aurora4 [25, 26] database was used to verify the effectiveness of the proposed approach for the medium vocabulary continuous speech recognition task. It contains speech data in the presence of additive noises and linear convolutional distortions, which were introduced synthetically to "clean" speech derived from WSJ [27] database. Two training sets were designed for this task. One is clean-condition training set consisting of 7138 utterances recorded by the primary Sennheiser microphone. The other one is multi-condition training set which is time-synchronized with the clean-condition training set. One half of the utterances were recorded by the primary Sennheiser microphone while the other half were recorded using one of a secondary microphone. Both halves include a combination of clean speech from clean-condition training set and speech corrupted by one of six different noises (street, train station, car, babble, restaurant, airport) at 10-20 dB SNR. These two training set pairs are also used for training DNN pre-processor. For evaluation, the original two sets consisted of 330 utterances from 8 speakers, which was recorded by the primary microphone and a secondary microphone, respectively. Each set was then corrupted by the same six noises used in the training set at 5-15 dB SNR, creating a total of 14 test sets. These 14 test sets were grouped into 4 subsets: clean (Set 1), noisy (Set 2 to Set 7), clean with channel distortion (Set 8), noisy with channel distortion (Set 9 to Set 14), which were denoted as A, B, C, and D, respectively.

Table 1: Performance (word error rate in %) comparison of GMM-HMM systems using MFCC features under different training conditions on the testing sets of Aurora4 databases.

System	A	B	C	D	Avg.
Clean-condition Training					
Noisy	8.0	36.7	23.7	52.1	40.3
DNN-PP	8.0	15.8	13.4	32.3	22.1
AFE	7.6	27.0	25.3	41.2	31.6
Multi-condition Training					
Noisy	12.5	17.6	19.3	31.0	23.1
DNN-PP	10.3	13.7	13.1	29.0	20.0
AFE	10.2	17.4	20.0	29.0	22.0

As for the front-end, the frame length was set to 25 msec with a frame shift of 10 msec for the 16kHz speech waveforms. Then 257-dimensional log-power spectra features were used to train DNN pre-processor. The DNN architecture was 1799-2048-2048-2048-257, which denoted that the sizes were 1799 (257*7, τ=3) for the input layer, 2048 for three hidden layers, and 257 for the output layer. Other parameter settings can refer to [24, 29]. Two acoustic feature types of ASR systems are adopted, namely 13-dimensional MFCC (including C_0) feature plus their first and second order derivatives, and 24-dimensional log Mel-filterbank feature plus their first and second order derivatives. Both MFCC and LMFB features are further processed by cepstral mean normalization.

For acoustic modeling, each triphone was modeled by a CDHMM with 3 emitting states. There were in total 3300 tied states based on decision trees. For GMM-HMM systems, each state had 16 Gaussian mixture components. A bigram language model (LM) for a 5k-word vocabulary was used in recognition. For DNN-HMM systems, the input layer was a context window of 11 frames of MFCC (11*39=429 units) or LMFB (11*72=792 units) feature vectors. All DNNs for acoustic modeling had 7 hidden layers with 2048 hidden units in each layer and the final soft-max output layer had 3296 units, corresponding to the tied stats of HMMs. The other parameters were set according to [15].

Table 1 gives a WER performance comparison of the GMM-HMM systems using MFCC features under different training conditions on the Aurora4 testing sets. For clean-condition training, our approach using DNN pre-processing (denoted as DNN-PP) achieved significant WER reductions on all test sets except the clean test set A, reducing the average WER from 40.3% to 22.1%. DNN-PP also outperformed advanced front-end (AFE) [30], with a relative WER reduction of 30.1%. For multi-condition training, with a much better baseline of 23.1% which was comparable to that of our approach in clean-condition training, our DNN-PP approach can still yield a remarkably relative WER reduction of 13.4% in average over the baseline, and 9.1% in average over AFE.

Table 2 lists a WER performance comparison of the DNN-HMM systems using the MFCC features. The baseline performance of the DNN-HMM systems in both clean-condition training and multi-condition training was improved by 12.4% and 39.0%, respectively, over the GMM-HMM systems in Table 1 which demonstrated the powerful capability of DNN-HMM and its noise robustness. In clean-condition training, our approach reduces the average WER from 35.3% to 18.7%, with a 47.0% relative improvement. In multi-condition training, with such a high baseline, our approach can further im-

Table 2: Word error rate (in %) comparison of DNN-HMM systems using MFCC features under different training conditions on the Aurora4 testing sets.

System	A	B	C	D	Avg.
Clean-condition Training					
Noisy	4.7	30.7	23.3	47.1	35.3
DNN-PP	5.1	12.0	10.5	29.0	18.7
Multi-condition Training					
Noisy	5.4	9.7	9.5	20.6	14.1
DNN-PP	4.9	8.3	8.2	20.6	13.3

Table 3: Performance (word error rate in %) comparison of DNN-HMM systems using LMFB features under different training conditions on the testing sets of Aurora4 databases.

System	A	B	C	D	Avg.
Clean-condition Training					
Noisy	4.2	30.8	22.5	47.6	35.5
DNN-PP	4.2	10.9	10.0	27.6	17.5
Multi-condition Training					
Noisy	4.6	8.4	7.8	18.6	12.5
DNN-PP	4.5	7.5	7.4	19.3	12.3

prove the performance for test sets A, B, and C. The reason why the performance of test set D was not improved might be that the DNN-based pre-processor could not well-learn the relationship between noisy and clean speech features when both additive noises and channel distortions were involved.

Table 3 shows a performance comparison of the DNN-HMM systems using the LMFB features. In clean-condition training, although the baseline performance was a little worse than that using the MFCC features, the performance after DNN pre-processing was the best compared with the corresponding results in Tables 1 and 2, which indicated that the LMFB features contained more useful speech information than the MFCC features. In multi-condition training, the baseline WER of 12.5% was the same as that reported in [23], which was the best baseline performance as far as we know. Furthermore our proposed approach could reduce the WER on top of this baseline, especially on the test set B. To our best knowledge, this is the first showcase of yielding performance gain by using an enhancement approach alone without adaptation for multi-condition training with log Mel-filterbank features and DNN acoustic modeling on the Aurora4 database.

In [23], it was claimed to have reported the best recognition results on the Aurora4 task with its proposed front-end by reducing the average WER from 15.3% to 14.2% . Compared with our proposed DNN-HMM systems based on MFCC features in multi-condition training without adaptation, we had reduced the average WER from 14.1% to 13.3%. Clearly, both our baseline and enhanced performances were better than the 14.2% WER reported in [23]. For the DNN-HMM systems based on the LMFB features, starting with the same average baseline results of 12.5%, the front-end presented in [23] even led to a WER increase to 14.3% while our proposed approach reduced the average WER to 12.3%.

Note that for Aurora4, the additive noise types and channel distortions of the test sets are exactly the same as those in the multi-condition training set, giving a well-matched training-testing condition. But in most real-world applications, it's difficult to obtain the noise information in advance. To simulate a more realistic scenario, we design a new multi-condition training set without knowing the noise information in the test sets, which included the clean speech utterances recorded by two microphones in the original multi-condition set, and noisy speech synthesized by adding 100 noise types [31] to the remaining utterances in the clean-condition set of Aurora4, at different SNRs from 0 dB to 15 dB with an increment of 5 dB, creating the final set of 7138 utterances.

Table 4: Performance (word error rate in %) comparison of DNN-HMM systems using LMFB features with a new multi-condition training set on the testing sets of Aurora4 databases.

System	A	B	C	D	Avg.
Clean-condition Training					
DNN-PP	4.3	20.1	10	37.2	25.6
New Multi-condition Training					
Noisy	4.7	11.4	9.7	25.1	16.7
DNN-PP	4.3	13.1	7.1	28.6	18.7

This new multi-condition training set was used for training of both front-end DNN (i.e., DNN pre-processor) and back-end DNN (i.e., DNN acoustic model). Table 4 gives a similar performance comparison as in Table 3 using the new multi-condition training set. The baseline performance of clean-condition training was the same as that in Table 3, which was not included in Table 4. In clean-condition training, DNN pre-processing trained with the new multi-condition training set still yielded a significant performance WER reduction from 35.5% to 25.6%. More interestingly, the baseline performance of the new multi-condition training could be even better than the best performance of clean-condition training in Table 3. These observations confirm that using multiple noise types for training of both front-end and back-end DNNs can well predict an unseen noise condition in the testing stage. For the new multi-condition training scenario, DNN pre-processing could not further improve the recognition performance on test sets B and D due to the mismatch of additive noise types between training and testing conditions while the WER was reduced on test sets A and C.

5. Conclusion and Future Work

We propose a DNN-based pre-processing framework for noise robust speech recognition. Contrary to traditional thinking, we demonstrate that promising results can be achieved by speech enhancement alone without any feature-based or model-based post-processing when tested on the Aurora4 ASR task. We have also shown that the proposed front-end produces better ASR results than competing pre-processors based on speech separation. Ongoing future work includes combining the proposed DNN-based preprocessing technique with other noise robust algorithms and focusing on how to further improve the performance for multi-condition training when both additive noises and convolutional distortion are involved in the test data. Approach to reducing potential mismatches in noise types between training and testing conditions will also be investigated.

6. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants No. 61305002 and the Programs for Science and Technology Development of Anhui Province, China under Grants No. 13Z02008-4 and No. 13Z02008-5.

7. References

[1] A. Acero, *Acoustic and Environment Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1993.

[2] Y. Gong, "Speech recognition in noisy environments: a survey," *Speech Communication*, Vol. 16, No. 3, pp. 261-291, 1995.

[3] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 22, No. 4, pp. 745-777, 2014.

[4] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora2 database," *Proc. EuroSpeech*, 2001, pp. 217-220.

[5] L. Buera, E. Lleida, A. Miguel, and A. Ortega, "Multi-environment models based linear normalization for robust speech recognition in car conditions," *Proc. ICASSP*, 2004, pp. 1013-1016.

[6] C. Cerisara and K. Daoudi, "Evaluation of the SPACE denoising algorithm on Aurora2," *Proc. ICASSP*, 2006, pp. I-521-I-524.

[7] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," *Proc. ICASSP*, 2007, pp. 377-380.

[8] J. Du, Y. Hu, L.-R. Dai, and R.-H. Wang, "HMM-based pseudo-clean speech synthesis for SPLICE algorithm," *Proc. ICASSP*, 2010, pp. 4570-4573.

[9] J. Du and Q. Huo, "Synthesized stereo-based stochastic mapping with data selection for robust speech recognition," *Proc. ISCSLP*, 2012, pp. 122-125.

[10] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, No. 5786, pp. 504-507, 2006.

[11] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, Vol. 18, pp. 1527-1554, 2006.

[12] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 20, No. 1, pp. 30-42, 2012.

[13] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 14-22, 2012.

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

[15] M. L. Seltzer, D. Yu, and Y.-Q. Wang, "An investigation of deep neural networks for noise robust speech recognition," *Proc. ICASSP*, 2013, pp. 7398-7402.

[16] B. Li, Y. Tsao, and K. C. Sim, "An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition," *Proc. INTERSPEECH*, 2013, pp. 3002-3006.

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

[18] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition," *Proc. ICASSP*, 2008, pp. 4041-4044.

[19] A. L. Maas, Q. V. Le, T. M. ONeil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," *Proc. INTERSPEECH*, 2012.

[20] J. Du, Y. Hu, L.-R. Dai, and R.-H.Wang, "Synthesized stereo mapping via deep neural networks for noisy speech recognition," *Proc. ICASSP*, 2014, pp. 1783-1787.

[21] W. Hartmann, A. Narayanan, E. Fosler-Lussier, and D.-L. Wang, "A direct masking approach to robust ASR," *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 21, No. 10, pp. 1993-2005, 2013.

[22] A. Narayanan and D.-L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," *Proc. ICASSP*, 2013, pp. 7092-7096.

[23] A. Narayanan and D.-L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 22, No. 4, pp. 826-835, 2014.

[24] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, Vol. 21, No. 1, pp. 65-68, 2014.

[25] H. G. Hirsch, *Experimental Framework for the Performance Evaluation of Speech Recognition Front-Ends on a Large Vocabulary Task, Version 2.0*, 2002.

[26] N. Parihar and J. Picone, *DSR Front End LVCSR Evaluation*, 2002.

[27] D. Paul and J. Baker, "The design of Wall Street Journal-based CSR corpus," *Proc. ICSLP*, 1992, pp. 899-902.

[28] S. Young *et al.*, The HTK Book (for HTK v3.4), 2006.

[29] G. Hinton, "A practical guide to training restricted Boltzmann machines," UTML TR 2010-003, University of Toronto, 2010.

[30] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithms*, ETSI ES 202 050 v1.1.1 (2002-10), Oct. 2002, ETSI standard document.

[31] G. Hu, 100 nonspeech environmental sounds, 2004. [http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html]