SYNTHESIZED STEREO MAPPING VIA DEEP NEURAL NETWORKS FOR NOISY SPEECH RECOGNITION

Jun Du[1], Li-Rong Dai[1], Qiang Huo[2]

[1]University of Science and Technology of China, Hefei, P. R. China
[2]Microsoft Research, Beijing, P. R. China
{jundu,lrdai}@ustc.edu.cn, qianghuo@microsoft.com

ABSTRACT

In our previous work, we extend the traditional stereo-based stochastic mapping by relaxing the constraint of stereo-data, which is not practical in real applications, via HMM-based speech synthesis to construct the "clean" channel data for noisy speech recognition. In this paper, we propose to use deep neural networks (DNNs) for stereo mapping compared with the joint Gaussian mixture model (GMM). The experimental results on Aurora3 databases show that our proposed DNN based synthesized stereo mapping can achieve consistently significant improvements of recognition performance over joint GMM based synthesized stereo mapping in the well-matched (WM) condition among four different European languages.

Index Terms— HMM-based speech synthesis, joint Gaussian mixture model, deep neural network, noisy speech recognition

1. INTRODUCTION

With the progress of automatic speech recognition (ASR), the noise robustness of speech recognizers attracts more and more attentions for practical recognition systems. Many techniques [17] 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 feature compensation approach by using stereo data to learn the mapping function between clean speech and noisy speech. SPLICE [12], namely stereo-based piecewise linear compensation for environments, is one successful showcase which is an extension of techniques [1, 24] developed at Carnegie Mellon University (CMU) in the past decades. Also similar approaches are proposed in [8, 9]. Recently, a stereo-based stochastic mapping technique[2, 3] is proposed, which outperforms SPLICE. The basic idea is to build a Gaussian mixture model (GMM) for the joint distribution of the clean and noisy speech by using stereo data. The simplicity to construct a joint GMM without environment selection makes it easier to implement in recognition stage.

One main problem of these approaches is the constraint of stereo data. Several works are presented to address this issue. In ([22, 29]), stochastic vector mapping (SVM), which represents the mapping from the noisy speech to clean speech by a simple transformation, is a generalized definition of SPLICE. And a joint training of the parameters of SVM function and HMMs is implemented by adopting maximum likelihood (ML) or minimum classification error (MCE) criteria. MMI-SPLICE [13] is much like SPLICE, but without the need for target clean features. Instead of learning a speech enhancement function, it learns to increase recognition accuracy directly with a maximum mutual information (MMI) objective function. FMPE [25], a kind of discriminatively trained features, is related with SPLICE to a certain extent [11].

The motivation of our approach is to relax the constraint of recorded stereo-data from a new viewpoint: synthesized pseudo-clean features generated by exploiting HMM-based synthesis method ([28, 30]) is used to replace the ideal clean features from one of the stereo channels in those stereo-based approaches. In [14], we demonstrate this approach can achieve even better performance than SPLICE in the clean training condition of Aurora2 database. In our recent work [16], we apply the synthesized features to stereo-based stochastic mapping approach with a data selection strategy, and further verify its effectiveness over a high-performance baseline of real-world ASR, namely the well-matched condition of Aurora3 databases.

Inspired by recent progress of deep learning [19], especially its application in speech recognition area ([10, 21, 26]), in this paper, we expand our previous work [16] by using deep neural network (DNN) for stereo mapping. It should be emphasized that here DNN is used for regression or function approximation, rather than more commonly used classification. Our experimental results on Aurora3 database show that DNN based synthesized stereo mapping can achieve very promising recognition accuracy. Compared with joint GMM used for stereo mapping in [16], DNN has the following advantages: 1) It can make full use of the acoustic context information via the neighbouring frames; 2) The singular problem of full covariance matrix estimation in joint GMM can be avoided; 3) The prediction is straightforward while there are always problems in minimum mean squared error (MMSE) estimation [16] or maximum a posterior (MAP) estimation [3] in the framework of probabilistic model. In terms of deep learning for regression problem, our work is related to recurrent neural network (RNN) based noise reduction for robust speech recognition [23], where stereo data of clean and noisy speech are used. Another relevant work is our most recent work on DNN based speech enhancement [31], where DNN also as a regression model provides better listening quality than other traditional approaches.

The remainder of the paper is organized as follows. In Section 2, we give a system overview of our proposed framework. In Section 3, we review joint GMM based approach and present our DNN based approach. In Section 4, we report experimental results and finally we conclude the paper in Section 5.

2. SYSTEM OVERVIEW

The overall flowchart of our propose framework is illustrated in Fig. 1. In the training stage, first a baseline system can be trained from multi-condition training data using MFCC features with cep-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

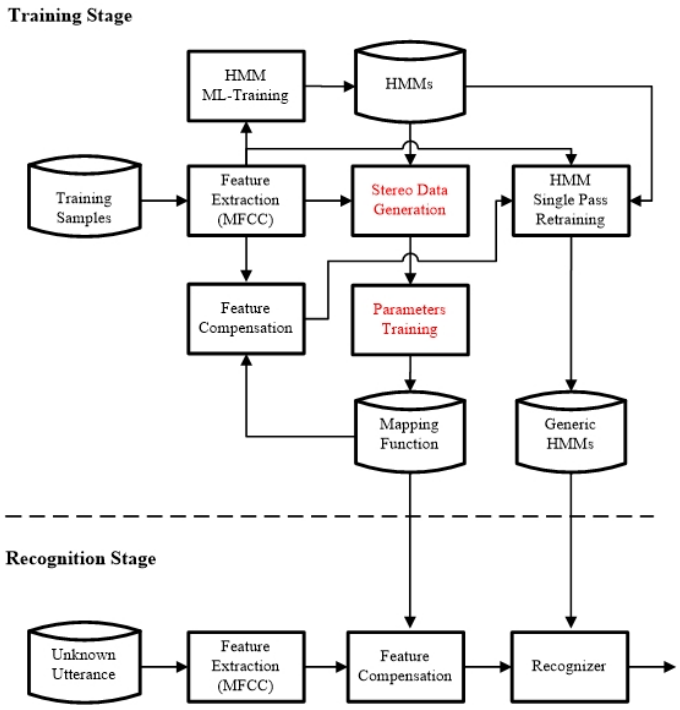

Fig. 1. Overall development flow and architecture.

stral mean normalization (CMN). Then the stereo feature vectors are generated via the training features and baseline HMMs, which are used to train the parameters of mapping functions. Followed by feature compensation to training features using the mapping function, generic HMMs are generated by using single pass retraining (SPR) [32], which is verified to be more effective than the retraining from the scratch [15]. The SPR works as follows: given one set of well-trained models, a new set matching a different training data parameterization 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 data and then switching to the new training data to compute the parameter estimation for the new set of models. In the recognition stage, after feature compensation to MFCC features extracted from the unknown utterance, the normal recognition is performed.

As for the stereo data generation module, suppose that we only have noisy speech as the training data in real applications. Then HMMs trained using those noisy features are noise-robust to some extent. To synthesize the features as the "clean" channel of the stereo data, first state-level forced-alignment of training features with true labels is performed. With this state sequence and corresponding HMMs, we can do the HMM-based speech synthesis [28]. The details of formulation can refer to [14]. Obviously, to the recognizer, those synthesized oracle feature sequences are perfectly matching and robust to not only noises, but also other irrelevant factors. In the following sections, we elaborate on two mapping functions, namely joint GMM and DNN.

3. STEREO MAPPING

3.1. Joint Gaussian Mixture Model

Assume we have a set of stereo data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, where $\boldsymbol{x}$ is the clean feature representation of speech, and $\boldsymbol{y}$ is the corresponding noisy feature representation. D is the dimension of feature vectors. Define $\boldsymbol{z} \equiv (\boldsymbol{x}, \boldsymbol{y})$ as the concatenation of the two channels. In the most general case, $\boldsymbol{y}$ representing L_n noisy speech vectors is used to predict $\boldsymbol{x}$ representing L_c clean speech vectors. To construct the mapping function between $\boldsymbol{y}$ and $\boldsymbol{x}$, the joint distribution $p(\boldsymbol{z})$ should be trained. Here Gaussian mixture model (GMM) is used:

$$p(\boldsymbol{z}) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{z,k}, \boldsymbol{\Sigma}_{zz,k}) \tag{1}$$

where K is the number of mixture components, ω_k, $\boldsymbol{\mu}_{z,k}$, and $\boldsymbol{\Sigma}_{zz,k}$, are the mixture weights, mean vector, and covariance matrix of each component, respectively. Then the mean vector $\boldsymbol{\mu}_{z,k}$ will be of dimension $D(L_c + L_n)$ and the covariance matrix $\boldsymbol{\Sigma}_{zz,k}$ will be of size $D(L_c + L_n) \times D(L_c + L_n)$.

The above joint GMM distribution can be estimated in a classical way using EM algorithm. In the feature compensation stage, minimum mean-squared error (MMSE) estimation is adopted:

$$\hat{\boldsymbol{x}} = E_x[\boldsymbol{x}|\boldsymbol{y}] = \sum_{k=1}^{K} P(k|\boldsymbol{y}) E_x[\boldsymbol{x}|\boldsymbol{y}, k] \tag{2}$$

where $P(k|\boldsymbol{y})$ is the posterior probability defined as

$$P(k|\boldsymbol{y}) = \frac{\omega_k \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{y,k}, \boldsymbol{\Sigma}_{yy,k})}{\sum_{k=1}^{K} \omega_k \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{y,k}, \boldsymbol{\Sigma}_{yy,k})} \tag{3}$$

and the conditional expectation $E_x[\boldsymbol{x}|\boldsymbol{y}, k]$ can be calculated as

$$E_x[\boldsymbol{x}|\boldsymbol{y}, k] = \boldsymbol{\mu}_{x,k} + \boldsymbol{\Sigma}_{xy,k} \boldsymbol{\Sigma}_{yy,k}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_{y,k}) \ . \tag{4}$$

In [2], it is indicated that the item $\boldsymbol{\Sigma}_{xy,k} \boldsymbol{\Sigma}_{yy,k}^{-1}$ in Eq.(4) represents the linear transformation to the noisy speech features. But according to the experiments of our proposed approach, we observe that this linear transformation can even result in poor recognition performance. One possible explanation is although the covariance parameters $\boldsymbol{\Sigma}_{xy,k}$ and $\boldsymbol{\Sigma}_{yy,k}$ trained under the maximum likelihood criterion for feature compensation in Eq.(2) can bring the minimum squared error between clean and noisy speech features, it may not necessarily improve the discriminations among classes of the speech recognizer. So in our implementation of feature compensation, Eq.(4) is modified as

$$E_x[\boldsymbol{x}|\boldsymbol{y}, k] = \boldsymbol{\mu}_{x,k} + (\boldsymbol{y} - \boldsymbol{\mu}_{y,k}) \tag{5}$$

which means only using bias compensation to noisy speech features is more stable than adding the linear transformation in this case. Another benefit from this modification is that we only need to train a joint GMM with diagonal covariance matrices, which can significantly reduce the number of model parameters. Acoustic context expansion by using several noisy feature vectors to predict the clean feature vector is another trick to improve the recognition performance [2], which increases the size of joint GMM. To achieve improvement of recognition performance but not increasing the size of joint GMM, we apply the following smoothing operation after feature compensation:

$$\hat{\boldsymbol{x}}_t^{\text{smooth}} = \frac{\sum_{\tau=-\Delta}^{\Delta} (\Delta + 1 - |\tau|) \hat{\boldsymbol{x}}_{t+\tau}}{\sum_{\tau=-\Delta}^{\Delta} (\Delta + 1 - |\tau|)} \tag{6}$$

where $\hat{\boldsymbol{x}}_t$ is the compensated feature vector of the t^{th} frame, and Δ is the size for context expansion. It is interesting that this simple operation plays a similar role to the acoustic context expansion method in [2] based on our experiments.

3.2. Deep Neural Network

A deep neural network (DNN) is a feed-forward, artificial neural network that has more than one layer of hidden units between its inputs and outputs [21]. In this work, DNN is adopted as a regression model to predict the clean features given the input noisy features with the acoustic context. The DNN training is illustrated in Fig. 2, which consists of generative pre-training and supervised fine-tuning.

The pre-training procedure treats each consecutive pair of layers as a restricted Boltzmann machine (RBM) [18] whose joint probability is defined as:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp\{-E(\boldsymbol{v}, \boldsymbol{h})\} \tag{7}$$

where $\boldsymbol{v}$ and $\boldsymbol{h}$ denote the observable variables and latent (hidden) variables, respectively. E is an energy function and Z is the partition function to ensure $p(\boldsymbol{v}, \boldsymbol{h})$ is a valid probability distribution. If both $\boldsymbol{v}$ and $\boldsymbol{h}$ are binary states, i.e., the Bernoulli-Bernoulli RBM, the energy function is given by

$$E(\boldsymbol{v}, \boldsymbol{h}) = -(\boldsymbol{b}_v^\top \boldsymbol{v} + \boldsymbol{b}_h^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W}_{vh} \boldsymbol{h}) \tag{8}$$

where $\boldsymbol{b}_v$, $\boldsymbol{b}_h$ are bias vectors of $\boldsymbol{v}$ and $\boldsymbol{h}$ respectively, and $\boldsymbol{W}_{vh}$ is the weight matrix between them. If $\boldsymbol{v}$ is real-valued data and $\boldsymbol{h}$ is binary, i.e., the Gaussian-Bernoulli RBM, the energy function is:

$$E(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{2}(\boldsymbol{v} - \boldsymbol{b}_v)^\top (\boldsymbol{v} - \boldsymbol{b}_v) - \boldsymbol{b}_h^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W}_{vh} \boldsymbol{h} \tag{9}$$

where we assume that the visible units follow the Gaussian noise model with an identity covariance matrix if the input data are pre-processed by mean and variance normalization.

The RBM parameters can be efficiently trained in an unsupervised fashion by maximizing the likelihood over training samples of visible units with the approximate contrastive divergence algorithm [18]. As for our DNN, a Gaussian-Bernoulli RBM is used for the first layer while a pile of Bernoulli-Bernoulli RBMs can be stacked behind the Gaussian-Bernoulli RBM. Then the parameters of RBMs can be trained layer-by-layer. Hinton *et al.* indicate that this greedy layer-wise unsupervised learning procedure always helps over the traditional random initialization.

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. We aim at minimizing mean squared error between the DNN output and the reference clean features:

$$E = \frac{1}{N} \sum_{n=1}^{N} \|(\hat{\boldsymbol{x}}_n(\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{b}) - \boldsymbol{x}_n)\|_2^2 \tag{10}$$

where $\hat{\boldsymbol{x}}_n$ and $\boldsymbol{x}_n$ are the n^{th} D-dimensional vectors of estimated and reference clean features, respectively. $\boldsymbol{y}_n$ is a $D(2L_{\text{w}} + 1)$-dimensional vector of input noisy feature with neighbouring left and right L_{w} frames as the acoustic context. $\boldsymbol{W}$ and $\boldsymbol{b}$ denote all the weight and bias parameters. The objective function is optimized using back-propagation procedure with conjugate gradient method in mini-batch mode of N sample frames.

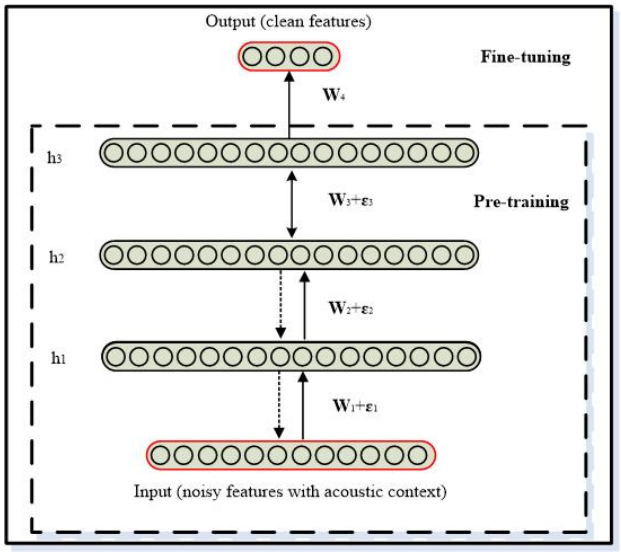

Fig. 2. Deep Neural Network.

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

In order to verify the effectiveness of the proposed approach on real-world ASR, Aurora3 databases are used, which contain utterances of digit strings recorded in real automobile environments for German, Danish, Finnish and Spanish, respectively. A full description of the above databases and the corresponding test frameworks are given in [4, 5, 6, 7].

In our ASR systems, each feature vector consists of 13 MFCCs (including C_0) plus their first and second order derivatives. The number of Mel-frequency filter banks is 23. MFCCs are computed based on power spectrum. CMN is applied to MFCC feature vectors. Each digit is modeled by a whole-word left-to-right CDHMM, which consists of 16 emitting states, each having 3 Gaussian mixture components. We focus on well-matched (WM) "training-testing" condition for experiments of Aurora3, where both training and testing data are recorded by close-talking (CT) and hands-free (HF) microphones. In all the experiments, tools in HTK [32] are used for training and testing. And the tools in [33] are used for generating the synthesized features.

The parameters for stereo mapping are set as follows. For joint GMM, $K = 4096$, $D = 13$, $L_c = L_n = 1$, $\Delta = 1$. The learning rate is set as 0.001 for RBM pre-training. And the number of epochs is 10 and 100 for RBM pre-training and DNN fine-tuning, respectively. The mini-batch size N is 100. Our other tuning parameters are set referring to [20]. Note that all the DNNs are trained separately for different languages.

4.2. Experimental Results

Fig. 3 gives a performance comparison of DNN based synthesized stereo mapping approach with different number of frames of input noisy feature with acoustic context information on the testing sets in the WM condition of Aurora3 Danish database. The configuration of 3 layers and 256 nodes for the hidden layer is used for DNN. We can observe that both too few frames (not enough context information) and too many frames (irrelevant information involved) can not result

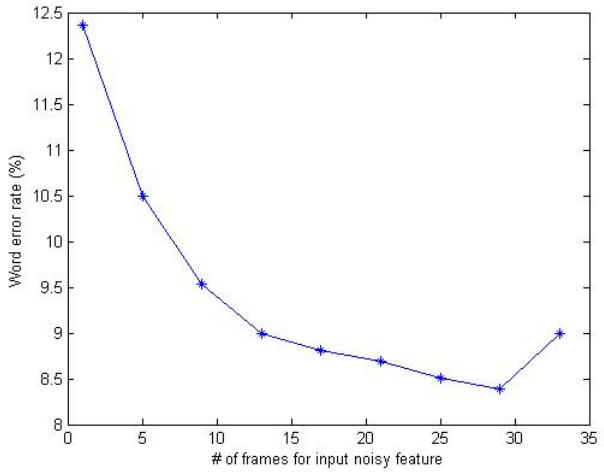

Fig. 3. Performance (word error rate in %) comparison of DNN based synthesized stereo mapping approach with different number of frames of input noisy feature on the testing sets in the WM condition of Aurora3 Danish database.

Table 1. Performance (word error rate in %) comparison of the baseline system and feature compensation systems using synthesized stereo mapping with different mapping functions on the testing sets in the WM condition of Aurora3 databases.

	German	Danish	Finnish	Spanish
Baseline	7.51	9.16	6.91	6.43
SSM-JGMM	6.59	8.00	5.44	5.87
SSM-DNN(3L)	6.25	6.93	3.53	5.42
SSM-DNN(4L)	5.89	6.04	2.57	4.12

in the best performance. In the extreme case that no acoustic context is used (1 frame), the word error rate is much higher than that of the baseline system. In our following experiments, the number of frames for input feature vectors is set as 29 (i.e., $L_{\mathrm{w}} = 14$), which is a good tradeoff used as the acoustic context.

Fig. 4 shows a performance comparison of DNN based synthesized stereo mapping approach with different nodes for the hidden layer of 3-layer neural network on the testing sets in the WM condition of Aurora3 databases. It is obvious that the recognition performance is consistently improved among four language with more hidden nodes until the number is 1024, which is set as default in the following DNN experiments. Further increasing the number of hidden nodes may lead to the over-fitting problem.

Table 1 compares the performance of the baseline system and feature compensation systems using synthesized stereo mapping with different mapping functions on the testing sets in the WM condition of Aurora3 databases. SSM-JGMM denotes the system using joint GMM based synthesized stereo mapping with data selection strategy which achieves the best results in [16]. SSM-DNN(3L) represents the system where DNN based synthesized stereo mapping is used with 3-layer neural network while SSM-DNN(4L) is corresponding to 4-layer neural network. First, all the feature compensation systems can achieve significant improvements of recognition performance compared with the baseline system. Furthermore, SSM-DNN systems can yield very significant error reduction over the SSM-JGMM system, especially on Finnish and

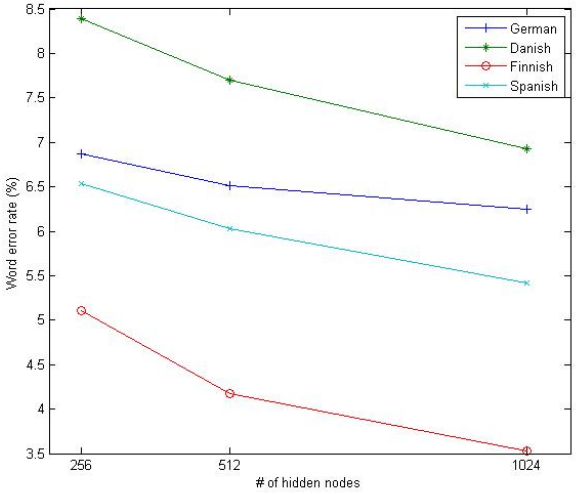

Fig. 4. Performance (word error rate in %) comparison of DNN based synthesized stereo mapping approach with different nodes for the hidden layer of 3-layer neural network on the testing sets in the WM condition of Aurora3 databases.

Spanish databases where more training data are provided for the deep learning. The reason why SSM-DNN is so effective may be explained as follows. On one hand, the input feature with acoustic context information can be fully utilized in the fully connected structure of deep neural network while joint GMM can not handle the dimension correlation and long-term context information very well. On the other hand, the prediction of clean speech features using DNN is straightforward while there are always problems in MMSE estimation or MAP estimation of joint GMM. In Table 1, we can also observe that SSM-DNN(4L) further reduces the error on the basis of SSM-DNN(3L) which indicates the effectiveness of the deep architecture. But due to the limitation of training data, the over-fitting problem occurs when using more than 4 layers neural network.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel feature compensation approach via DNN based synthesized stereo mapping for noisy speech recognition where "clean" channel data is generated by HMM-based speech synthesis. Our experiments on a real-world in-vehicle connected digits recognition task on Aurora3 benchmark databases show that for synthesized stereo mapping, DNN based approach can achieve very significant error reductions over the joint GMM based approach. Ongoing and future works include 1) to give a performance comparison between our feature compensation approach and noise robust DNN-HMM system reported in [27], and 2) to verify its effectiveness on large vocabulary speech recognition tasks.

6. ACKNOWLEDGMENT

This work was supported in part 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] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," *Proc. ICASSP*, 2007, pp.377-380.

[3] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 17, No. 7, pp.1325-1334, 2009.

[4] Aurora document AU/217/99, "Availability of Finnish SpeechDat-Car database for ETSI STQ WI008 front-end standardisation," Nokia, Nov. 1999.

[5] Aurora document AU/271/00, "Spanish SDC-Aurora database for ETSI STQ Aurora WI008 advanced DSR front-end evaluation: description and baseline results," UPC, Nov. 2000.

[6] Aurora document AU/273/00, "Description and baseline results for the subset of the SpeechDat-Car German database used for ETSI STQ Aurora WI008 advanced DSR front-end evaluation," Texas Instruments, Dec. 2001.

[7] Aurora document AU/378/01, "Danish SpeechDat-Car digits database for ETSI STQ-Aurora advanced DSR," Aalborg University, Jan. 2001.

[8] 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.

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

[10] 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.

[11] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Process. Lett.*, Vol. 12, No. 6, pp.477-480, 2005.

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

[13] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," *Proc. EuroSpeech*, 2005, pp.989-992.

[14] 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.

[15] J. Du and Q. Huo, "IVN-based joint training of GMM and HMMs using an improved VTS-based feature compensation for noisy speech recognition," *Proc. INTERSPEECH*, 2012.

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

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

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

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

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

[21] 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.

[22] Q. Huo and D.-L. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," *Proc. ICSLP*, 2006, pp.1129-1132.

[23] 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.

[24] P. J. Moreno, *Speech recognition in noisy environments*, Ph.D. thesis, Carnegie Mellon University, 1996.

[25] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," *Proc. ICASSP*, 2005, pp.961-964.

[26] F. Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," *Proc. INTERSPEECH*, 2011, pp.437-440.

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

[28] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proc. ICASSP*, 2000, pp.1315-1318.

[29] J. Wu and Q. Huo, "An environment-compensated minimum classification error training approach based on stochastic vector mapping," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14, No. 6, pp.2147-2155, 2006.

[30] Z.-J. Yan, F. K. Soong, and R.-H. Wang, "Word graph based feature enhancement for noisy speech recognition," *Proc. ICASSP*, 2007, pp.373-376.

[31] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," Accepted by Signal Processing Letter.

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

[33] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," *ISCA Workshop on Speech Synthesis*, 2007, pp.294-299.