DENOISING DEEP NEURAL NETWORKS BASED VOICE ACTIVITY DETECTION

Xiao-Lei Zhang and Ji Wu

Multimedia Signal and Intelligent Information Processing Laboratory,
Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing, China.
huoshan6@126.com, wuji_ee@tsinghua.edu.cn

ABSTRACT

Recently, the deep-belief-networks (DBN) based voice activity detection (VAD) has been proposed. It is powerful in fusing the advantages of multiple features, and achieves the state-of-the-art performance. However, the deep layers of the DBN-based VAD do not show an apparent superiority to the shallower layers. In this paper, we propose a denoising-deep-neural-network (DDNN) based VAD to address the aforementioned problem. Specifically, we pre-train a deep neural network in a special unsupervised denoising greedy layer-wise mode, and then fine-tune the whole network in a supervised way by the common back-propagation algorithm. In the pre-training phase, we take the noisy speech signals as the visible layer and try to extract a new feature that minimizes the reconstruction cross-entropy loss between the noisy speech signals and its corresponding clean speech signals. Experimental results show that the proposed DDNN-based VAD not only outperforms the DBN-based VAD but also shows an apparent performance improvement of the deep layers over shallower layers.

Index Terms— Deep learning, denoising deep neural networks, voice activity detection.

1. INTRODUCTION

Voice activity detectors (VADs) help to separate speech from its background noises. They are important frontends of modern speech processing systems, such as speech recognition systems [1–3] and speech communication systems [4]. Recently, the machine-learning-based VADs have received much attention in that they have the following notable merits. First, they can be integrated to the speech recognition systems naturally. Second, they have strong theoretical bases that guarantee the performance. Third, they can fuse the advantages of multiple features [5–9] much better than traditional VADs.

The machine-learning-based VADs can be categorized to four groups [10–16]. The first group is the discriminative-weight-training-based VADs [10, 12, 16]. They conduct linear weighted combinations of multiple features in the original feature space. The second group is the support-vector-machine (SVM) based VADs [11,13]. They first fuse multiple features to a long feature vector in the original feature space, and then project the long feature vector to the kernel-induced feature space for better classification performance. The third group is the multiple-kernel-SVM (MK-SVM) based VAD [14, 15]. It takes the distribution diversity of multiple features into consideration by first projecting different features into different kernel spaces and then fuse the features in the kernel spaces in a way with linear weighted combination. All of the aforementioned three groups utilize shallow models, i.e. models with only zero or one hidden layer, which lack the ability of describing highly variant features and discovering the underlying manifold of the features.

The fourth group is the deep-belief-networks (DBNs) based VAD [17]. Fundamentally, because the DBN [18] contains multiple hidden layers, the DBN-based VAD can describe highly variant features; because the unsupervised pre-training phase of DBN provides an initial point that is close to a good solution, the DBN-based VAD has a strong generalization ability when compared with other machine-learning-based VADs. However, the deep layers of the DBN-based VAD do not yield an apparent superiority to the shallower layers. In our personal opinion, it might not be proper to simply consider VAD as a binary-class classification problem with the noisy speech and the background noise as the two classes, since the background noise also contributes to the distribution of the noisy speech. This might account for the inapparent superiority of the deep layers over shallower layers in the DBN-based VAD.

In this paper, we propose a novel denoising deep-neural-networks (DDNNs) based VAD. The DDNN training also

This work is supported in part by the National Natural Science Funds of China under Grant 61170197, in part by the China Postdoctoral Science Foundation funded project under Grant 2012M520278, in part by the Planned Science and Technology Project of Tsinghua University under Grant 20111081023, and in part by the subproject of the National High-Tech. R&D Key Project of China (863 Key Project) under Grant 2012AA011004.

consists of two phases. The first phase is a special unsupervised denoising greedy layer-wise pre-training phase. The pre-training process of each hidden layer tries to extract a new feature that minimizes the *reconstruction cross-entropy loss* between the noisy speech signals and its corresponding clean speech signals (but not the noisy speech signals). The second phase is the well-known supervised fine-tuning phase. It groups all layers with the pre-trained parameters to a whole deep neural networks and tune the parameters for the minimum classification error. Experimental results show that the proposed DDNN-based VAD not only outperforms the DBN-based VAD but also shows an apparent performance improvement of the deep layers over shallower layers.

2. DENOISING-DNN-BASED VAD

The training process of the DDNN-based VAD consists of two phases – unsupervised denoising layer-wise pre-training phase and supervised fine-tuning phase, which are presented in detail in Sections 2.1 and 2.2 respectively. The overview of the DDNN-based VAD is presented in Algorithm 1.

Algorithm 1 Denoising-DNN-based VAD.

Input: Feature set $\left\{\mathbf{x}_i^{(0)}, \tilde{\mathbf{x}}_i^{(0)}, y_i^{(0)}\right\}_{i=1}^{n}$, the depth of the DDNN L

Output: Feature extraction model $\left\{\theta^{(l)}\right\}_{l=1}^{L}$, and the linear classifier above the model.

```
/* Unsupervised denoising layer-wise pre-training */
for l = 1, ..., L do
    Get θ^(l) by solving J_{θ^(l),θ'^(l)}(x^(l-1); x̃^(l-1)) defined in equation (1)
    Calculate x^(l-1) by equation (2)
    if l > 1 then
        Get θ̃^(l-1) by solving J_{θ̃^(l-1),θ̃'^(l-1)}(x̃^(l-2); x̃^(l-2)) or by the contrastive divergence learning [19].
        Calculate x̃^(l-1) by equation (3)
    end if
end for
/* Supervised fine-tuning */
Construct the classification-DDNN and fine-tune it by the back-propagation algorithm for the minimum classification error mentioned in Section 2.2.
```

2.1. Unsupervised Denoising Layer-wise Pre-training

Suppose we have D-dimensional noisy speech observations (i.e. frames) $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$ and their corresponding clean speech observations $\{\tilde{\mathbf{x}}_i, y_i\}_{i=1}^{n}$ with $\mathbf{x}_i = [x_{i,d}]_{d=1}^{D}$, $y_i \in \{H_0, H_1\}$, where $x_d \in [0,1]$ and H_1/H_0 denote the speech/noise hypothesis.

The layer-wise pre-training of each module of DDNN consists of optimizing two activation functions jointly. The first function, denoted as $f_\theta(\cdot)$, maps the noisy speech observation from the visible layer $\mathbf{x}$ to a hidden layer $f_\theta(\mathbf{x})$. The second function, denoted as $g_{\theta'}(\cdot)$, tries to reconstruct $\tilde{\mathbf{x}}$ (but not $\mathbf{x}$) from the hidden layer by $g_{\theta'}\left(f_\theta(\mathbf{x})\right)$.

The unsupervised pre-training tries to minimize the reconstruction cross-entropy loss between $\{\mathbf{x}_i\}_{i=1}^{n}$ and $\{\tilde{\mathbf{x}}_i\}_{i=1}^{n}$ which is defined as follows

$$\mathcal{J}_{\theta,\theta'}(\mathbf{x};\tilde{\mathbf{x}}) = \min_{\theta,\theta'} \sum_{i=1}^{n} L\left(\tilde{\mathbf{x}}_i; g_{\theta'}\left(f_\theta\left(\mathbf{x}_i\right)\right)\right) \tag{1}$$

with $L\left(\mathbf{x}_i;\mathbf{z}_i\right)$ defined as

$$L\left(\mathbf{x}_i;\mathbf{z}_i\right) = -\sum_{d=1}^{D}\left(x_{i,d}\log z_{i,d} + (1-x_{i,d})\log(1-z_{i,d})\right)$$

where $\mathbf{z}_i$ is short for $g_{\theta'}\left(f_\theta\left(\mathbf{x}_i\right)\right)$. Problem (1) can be solved locally by the well-known back-propagation algorithm.

When we try to pre-train the L-th module with $L > 1$ (i.e. the module is not the lowest one), we should first construct its input layer $\mathbf{x}^{(L-1)}$ by transferring $\mathbf{x}^{(0)}$ through the pre-trained networks as follows

$$\mathbf{x}^{(L-1)} = f_{\theta^{(L-1)}}\left(\ldots f_{\theta^{(l)}}\left(\ldots f_{\theta^{(2)}}\left(f_{\theta^{(1)}}\left(\mathbf{x}^{(0)}\right)\right)\right)\right) \tag{2}$$

where l denotes the l-th hidden layer (i.e. the l-th layer-wise module from the bottom-up), and $\mathbf{x}^{(0)}$ is the original feature vector.

Here comes the question. What should $\mathbf{x}^{(L-1)}$ reconstruct? Here, we propose to pre-train a clean-speech to clean-speech deep network that accompanies with the noisy-signal to clean-signal deep network, so that we can get $\tilde{\mathbf{x}}^{(l-1)}$ by

$$\tilde{\mathbf{x}}^{(L-1)} = f_{\tilde{\theta}^{(L-1)}}\left(\ldots f_{\tilde{\theta}^{(l)}}\left(\ldots f_{\tilde{\theta}^{(2)}}\left(f_{\tilde{\theta}^{(1)}}\left(\tilde{\mathbf{x}}^{(0)}\right)\right)\right)\right) \tag{3}$$

There are two ways to pre-train the accompanying deep network $\{f_{\tilde{\theta}^{(l)}}\}_{l=1}^{L-1}$ (i.e. the deep neural network for the clean-speech-to-clean-speech reconstruction) in the layer-wise greedy training mode. The first one is to minimize the reconstruction cross-entropy loss via (1) with $\tilde{\mathbf{x}}$ as both the input and the target of the module. Another way is to maximize the *logrithmic likelihood* of $\tilde{\mathbf{x}}$ by the efficient *contrastive divergence* algorithm proposed in DBN [19]. In this paper, we adopt the former for simplicity. Note that we cannot use $\mathbf{x}^{(L-1)}$ to recover $\tilde{\mathbf{x}}^{(0)}$ directly for saving the computation load of constructing $\tilde{\mathbf{x}}^{(L-1)}$, since it's unlikely to describe the extraction network $\{f_{\theta^{(l)}}\}_{l=1}^{L-1}$ of the noisy speech simply by a single hidden-layer reconstruction network $g_{\theta'^{(1)}}$.

In this paper, all activation functions $f_{\theta^{(l)}}\left(\mathbf{x}^{(l-1)}\right)$ and $g_{\theta'^{(l)}}\left(\hat{\mathbf{x}}^{(l-1)}\right)$ are defined as $f_{\theta^{(l)}}\left(\mathbf{x}^{(l-1)}\right) = s\left(\mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}\right)$ and $g_{\theta'^{(l)}}\left(\hat{\mathbf{x}}^{(l-1)}\right) = s\left(\mathbf{W}'^{(l)}\hat{\mathbf{x}}^{(l-1)} + \mathbf{b}'^{(l)}\right)$ respectively with the function $s(x)$ set to the logistic function $s(x) = 1/(1+e^{-x})$ and $\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$ denoted as the weight matrix and the bias term between the $(l-1)$-th and l-th layers of the network respectively.

2.2. Supervised Fine-tuning

The supervised fine-tuning phase can be divided into three steps. The first step is to construct the feature extraction part of the DDNN by first discarding the function $\{g_{\theta'^{(l)}}\}_{l=1}^{L}$ and the accompanying deep networks $\{f_{\tilde{\theta}}, g_{\tilde{\theta}'}\}_{l=1}^{L-1}$ and then stacking all pre-trained functions $\{f_{\theta^{(l)}}\}_{l=1}^{L}$ layer by layer as [18] did. The second step is to add a linear classifier above the feature extraction part so as to formulate the entire DDNN. The third step is to fine-tune DDNN by the common back-propagation algorithm for the minimum classification error (MCE), where the cross-entropy loss is also used as the surrogate relaxation function. We call the DDNN for MCE as the classification-DDNN. Note that another usage of DDNN is to only carry out the first step of the classification-DDNN, and then take the extracted denoising features as the input of some independent classifiers, such as SVM. We call the DDNN for extracting denoising features as the reconstruction-DDNN. We only consider the classification-DDNN in this paper.

3. MOTIVATION AND RELATED WORK

The proposed algorithm can be viewed as an idea combination of the stacked denoising autoencoder (SDAE) [20, 21] and speech enhancement techniques [22]. SDAE, proposed by Vincent *et al.* in 2008 [20, 21], is a novel deep learning technique that has shown comparable performance with DBN. It first adds noise to the original clean features and then takes the noisy features as the input of the module that is to be pre-trained. But it does not try to reconstruct the noisy features. Instead, it tries to recover the original clean features by minimizing the cross-entropy loss or the squared error loss between the reconstructed features and the original clean features. Compared with SDAE, DDNN also tries to recover the clean features, but the noise injected to the clean features is from the real environment instead of from artificial addition.

Speech enhancement techniques, such as the minimum mean square error estimation [22], try to estimate the amplitude of the clean speech from the noisy speech observation, which is also known as the *a priori* signal-to-noise ratio (SNR) estimation. The speech enhancement techniques have been widely employed in the VAD research, such as the well-known Sohn VAD [23]. Compared with the speech enhancement techniques, we construct a deep architecture in a machine-learning perspective for the clean speech estimation with an assumption that the training data has its corresponding clean speech target, while some speech-enhancement-based VADs assume that the background noise is relatively stationary, so that they can trust the statistical parameters updated in the silence period for the clean speech estimation when the speech activity appears. We have to note that many speech enhancement techniques do not need the silence period for the noise spectrum estimation, such as [24].

4. EXPERIMENTS

Seven noisy test corpora of AURORA2 are used for performance analysis. Four signal-to-noise ratio (SNR) levels of the audio signals are selected, which are $[-5, 0, 5, 10]$dB respectively. Each test corpus of AURORA2 contains 1001 utterances, which are split randomly into three groups for training, developing and test respectively. Each training set and development set consist of 300 utterances respectively. Each test set consists of 401 utterances. Note that the corpora in the same background noise scenario but at different SNR levels are split with the same random seed, and have the same manual labels. We concatenate all short utterances in each data set to a long one so as to simulate the real-world application environment of VAD. Eventually, the length of each long utterance is in a range of (450,750)s long with the percentages of speech ranging from 54.57% to 73.32%.

The sampling rate is 8kHz. We set the frame length to 25ms long with a frame-shift of 10ms. We extract 10 acoustic features from each observation. The detailed information of the features are listed in Table 1. All features are normalized into the range of $[0, 1]$ in dimension.

Table 1. Features and their attributes. The subscript of each feature is the window length of the feature [25].

ID	Feature	Dimension	ID	Feature	Dimension
1	Pitch	1	7	MFCC_{16}	20
2	DFT	16	8	LPC	12
3	DFT_8	16	9	RASTA-PLP	17
4	DFT_{16}	16	10	AMS	135
5	MFCC	20		**Total**	273
6	MFCC_8	20			

The SVM-based VAD, MK-SVM-based VAD, and DBN-based VAD are used for comparison. For the SVM-based VAD, DBN-based VAD, and DDNN-based VAD, we concatenate all 10 features in serial to a long feature vector and take the long feature vector as the input of the classifiers. For the MK-SVM-based VAD, we deal with the features in a similar way with [26].

In respect of the parameter setting, for the SVM-based and MK-SVM-based VADs, the Gaussian RBF kernel is used. The parameters of SVM and MK-SVM are searched in grid. For the DBN-based and DDNN-based VADs, up to three hidden layers are adopted with the numbers of the hidden units set to $[54, 7, 7]$ respectively. The learning rate of the unsupervised pre-training is set to 0.004. The maximum epoch of the unsupervised pre-training is set to 200. The learning rate of the supervised fune-tuning is set to 0.005. The maximum epoch of the supervised fune-tuning is set to 130. The batch mode training is adopted. Each batch contains 512 observations. Note that the parameters are selected empirically for a compromise between the training time complexity and

Table 2. Accuracy comparison in the babble, car, restaurant, and street noises. The subscripts of the DBN and DDNN are the depths (i.e. the numbers of the hidden layers) of the deep neural networks.

	Babble				Car				Restaurant				Street			
	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB
SVM	54.61	64.46	75.97	79.53	72.20	81.59	86.34	87.60	69.04	74.22	82.09	84.83	58.32	67.98	74.88	78.12
MKSVM	55.43	65.02	76.17	80.18	75.01	83.50	86.38	87.94	70.44	75.71	83.25	86.30	63.38	73.35	77.60	79.10
DBN_1	**61.03**	69.01	78.83	80.99	77.24	84.10	**87.18**	**88.48**	**70.23**	**75.73**	83.43	**86.12**	66.63	73.15	78.47	80.42
DBN_2	60.81	69.24	78.94	**81.23**	**77.88**	**84.14**	87.04	88.44	70.10	75.68	**83.59**	86.08	**67.41**	**73.76**	78.70	**80.86**
DBN_3	60.55	**69.38**	**79.03**	80.78	77.75	83.97	87.00	88.14	69.75	75.57	83.54	85.92	67.33	72.83	**79.03**	80.49
$DDNN_1$	**60.69**	69.42	78.61	81.39	76.06	83.86	86.77	88.17	**69.76**	75.88	83.47	86.41	66.21	72.21	79.33	81.24
$DDNN_2$	58.62	69.07	78.85	81.62	76.80	84.04	86.96	88.54	69.71	76.05	**83.90**	86.62	65.51	72.72	79.17	81.53
$DDNN_3$	57.84	**69.61**	**79.14**	**81.65**	**76.82**	**84.22**	**87.09**	**88.67**	69.55	**76.04**	83.78	**86.65**	**65.89**	**72.82**	**79.47**	**81.71**

Table 3. Accuracy comparison in the airport, train, and subway noises. "AVR" is short for average. "ALL" denotes that the AVR is calculated over all noise types and SNR levels. Note that when we calculate the averages, we did not consider the results of the babble noise in -5 and 0 dB, since the manifolds of the speech and background noise are similar in that situation.

	Airport				Train				Subway				**AVR over diff. noise types**				**AVR**
	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB	-5dB	0dB	5dB	10dB	ALL
SVM	64.48	74.26	80.94	85.21	66.24	74.29	82.91	85.28	74.75	81.24	83.58	85.18	67.51	75.60	80.96	83.68	76.93
MKSVM	65.86	75.59	82.30	85.38	68.78	76.31	83.99	85.34	79.90	84.82	86.11	87.46	70.56	78.21	82.26	84.53	78.89
DBN_1	66.18	76.63	81.89	**86.63**	68.59	**76.95**	**83.65**	**85.72**	78.54	82.70	85.60	85.79	71.24	78.21	82.72	84.88	79.26
DBN_2	66.35	**76.66**	**81.92**	86.41	**68.99**	**76.95**	83.49	85.68	**79.10**	**83.29**	85.77	**86.25**	**71.64**	**78.41**	82.78	**84.99**	**79.46**
DBN_3	**66.62**	76.38	81.85	86.50	68.89	76.14	83.56	85.62	78.95	83.26	**85.81**	86.01	71.55	78.03	**82.83**	84.78	79.30
$DDNN_1$	66.00	76.61	82.34	86.81	68.59	77.36	83.88	85.94	77.90	83.20	**85.84**	**86.64**	70.75	78.19	82.89	85.23	79.27
$DDNN_2$	66.80	76.86	**82.45**	**86.98**	69.33	77.48	84.21	86.12	78.19	83.39	85.62	86.46	71.06	78.42	83.02	85.41	79.48
$DDNN_3$	**67.00**	**76.85**	82.30	86.85	**69.44**	**77.60**	**84.25**	**86.16**	**78.53**	**83.60**	85.73	86.49	**71.21**	**78.52**	**83.11**	**85.45**	**79.57**

the accuracy. We run all experiments 10 times and report the average performances. The reported performance might be further improved by tuning the parameters.

Tables 2 and 3 list the experimental results. The highlighted contents of each column are the best performance of the referenced DBN-based VAD and that of the DDNN-based VAD on the corresponding noise scenario respectively. From the two tables, we can see that the deep layers of the DDNN-based VADs perform better than the shallower layers, which supports our conjecture in Section 3. Also, the DDNN-based VAD outperforms the SVM-based VAD and the MK-SVM-based VAD. Moreover, the DDNN-based VAD even outperforms the DBN-based VAD in several noise scenarios, which demonstrates its effectiveness. The experimental phenomenon manifested our conjecture in the introduction section about the reason why the deep layers the DBN-based VAD does not outperform the shallow layers. That is, the manifolds of the clean speech and background noise mixed with each other, so that we cannot expect DBN to distinguish the background noise from the noisy speech that contains the manifolds of both the clean speech and the background noise.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a denoising-deep-neural-networks-based VAD. Specifically, the DDNN training contains two phases. The first phase is to pre-train a deep neural network in an unsupervised denoising greedy layer-wise mode. The second phase is to fine-tune the whole deep neural network as usual. The denoising pre-training makes the DDNN discover the manifold of the clean speech without suffering severely from the disruption of the background noise. Experimental results have shown that the deep layers of the DDNN-based VAD are much more powerful than the shallower layers, and moreover, the DDNN-based VAD outperforms the DBN-based VAD in several noise scenarios.

However, to train a DDNN model, the noisy speech training corpus needs its corresponding clean corpus, which is an ideal situation. Therefore, how to relax this constraint is what we focus on in the future work. Moreover, the experiments are limited to the matching environments, how to make the DDNN-based VAD perform steadily in unmatching environments is another key problem we want to address.

Acknowledgment: The authors would like to thank the anonymous referees for their valuable advice, which greatly improved the quality of this paper.

6. REFERENCES

[1] D. Yu and L. Deng, "Deep-structured hidden conditional random fields for phonetic recognition," in *Proc. INTERSPEECH*, 2010, pp. 2986–2989.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 11, pp. 2–17, 2012.

[4] K. Hana and D. L. Wang, "A classification based approach to speech segregation," *The Journal of the Acoustical Society of America*, vol. 99, pp. 1–34, 2012.

[5] D. L. Wang, "The time dimension for scene analysis," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1401–1426, 2005.

[6] D. L. Wang and G. J. Brown, *Computational auditory scene analysis: principles, algorithms and applications*, Wiley-IEEE Press, 2006.

[7] Y. X. Wang, K. Han, and D. L. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 1, no. 99, pp. 1–10, 2012.

[8] Y. X. Wang and D. L. Wang, "Cocktail party processing via structured prediction," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1–8.

[9] Y. X. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. PP, no. 99, pp. 1–23, 2013.

[10] S. I. Kang, Q. H. Jo, and J. H. Chang, "Discriminative weight training for a statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 15, pp. 170–173, 2008.

[11] J. W. Shin, J. H. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Language*, vol. 24, no. 3, pp. 515–530, 2010.

[12] T. Yu and J. H. L. Hansen, "Discriminative training for multiple observation likelihood ratio based voice activity detection," *IEEE Signal Process. Lett.*, vol. 17, no. 11, pp. 897–900, 2010.

[13] J. Wu and X. L. Zhang, "Maximum margin clustering based statistical VAD with multiple observation compound feature," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 283–286, 2011.

[14] J. Wu and X. L. Zhang, "Efficient multiple kernel support vector machine based voice activity detection," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 466–499, 2011.

[15] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 1, no. 99, pp. 1–24, 2012.

[16] Y. Suh and H. Kim, "Multiple acoustic model-based discriminative likelihood ratio weighting for voice activity detection," *IEEE Signal Process. Lett.*, vol. 19, no. 8, pp. 507–510, 2012.

[17] X. L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 3371–3408, 2013.

[18] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2005, pp. 17–25.

[20] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[22] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoustic, Speech, Signal Process.*, vol. 32, no. 6, pp. 1109–1121, 1984.

[23] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, 1999.

[24] Israel Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 5, pp. 466–475, 2003.

[25] J. Ramírez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 689–692, 2005.

[26] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1175–1182.