Zhang *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2015) 2015:12
DOI 10.1186/s13636-015-0056-7

EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification

Zhaofeng Zhang[1], Longbiao Wang[1*], Atsuhiko Kai[2], Takanori Yamada[2], Weifeng Li[3] and Masahiro Iwahashi[1]

Abstract

Deep neural network (DNN)-based approaches have been shown to be effective in many automatic speech recognition systems. However, few works have focused on DNNs for distant-talking speaker recognition. In this study, a bottleneck feature derived from a DNN and a cepstral domain denoising autoencoder (DAE)-based dereverberation are presented for distant-talking speaker identification, and a combination of these two approaches is proposed. For the DNN-based bottleneck feature, we noted that DNNs can transform the reverberant speech feature to a new feature space with greater discriminative classification ability for distant-talking speaker recognition. Conversely, cepstral domain DAE-based dereverberation tries to suppress the reverberation by mapping the cepstrum of reverberant speech to that of clean speech with the expectation of improving the performance of distant-talking speaker recognition. Since the DNN-based discriminant bottleneck feature and DAE-based dereverberation have a strong complementary nature, the combination of these two methods is expected to be very effective for distant-talking speaker identification. A speaker identification experiment was performed on a distant-talking speech set, with reverberant environments differing from the training environments. In suppressing late reverberation, our method outperformed some state-of-the-art dereverberation approaches such as the multichannel least mean squares (MCLMS). Compared with the MCLMS, we obtained a reduction in relative error rates of 21.4% for the bottleneck feature and 47.0% for the autoencoder feature. Moreover, the combination of likelihoods of the DNN-based bottleneck feature and DAE-based dereverberation further improved the performance.

Keywords: Speaker recognition; Bottleneck features; Denoising autoencoder; Deep neural network; Reverberant speech

1 Introduction

Although speaker recognition has been researched for many years, most applications still require a microphone located near the speaker. However, many applications would benefit from speaker recognition through distant-talking speech capture, where the speaker is able to speak at some distance from the microphones. While in this task, even in quiet conditions, the microphone records not only the direct sound of the specific speaker but also reverberation signals. A reverberation signal is created when a sound or signal is reflected, causing a large number of reflections to build up and then decay as the sound is absorbed by the surfaces of objects in the space, which could include walls, furniture, people, and air.

Owing to the effects of reverberation, the accuracy of distant-talking speaker identification is significantly reduced. According to [1], approaches for dealing with reverberation can be classified as front-end- or back-end-based approaches. Approaches of the former type attempt to reduce the effect of reverberation from the observed speech signal [2-5], while the latter methods attempt to modify the acoustic model and/or decoder to suit a reverberant environment [6,7]. In this paper, we focus

*Correspondence: wang@vos.nagaokaut.ac.jp
[1]Nagaoka University of Technology, 1603-1 Kamitomioka-cho, Nagaoka, Niigata, 940-2188, Japan
Full list of author information is available at the end of the article

on front-end-based approaches for distant-talking speaker identification.

Many front-end-based techniques have been proposed for robust automatic speech recognition (ASR) and speaker recognition in distant-talking environments [2,4,5,8-18].

Cepstral mean normalization (CMN) [19-22] is considered the most general approach for dereverberation. However, the length of an impulse response in a distant-talking environment is usually much longer than the size of the analysis window in short-term spectral analysis. Therefore, CMN cannot compensate for late reverberation. Several studies have focused on mitigating this problem [4,5,13,17,23].

Beamforming [8,24], which is a simple and robust means of spatial filtering, can be used to suppress any signal from noise or the direction of reflection; therefore, it is effective for dereverberation [13,25]. Recently, a two-stage beamforming approach [26] was presented for dereverberation and noise reduction. The first stage comprises a delay-and-sum beamformer that generates a reference signal containing a spatially filtered version of the desired speech and the interference. The second stage uses the filtered microphone signals and the noisy reference signal to estimate the desired speech. However, good performance cannot be achieved, particularly when the reverberation is very strong.

In [27,28], a method based on mean subtraction using a long-term spectral analysis window was proposed. The results showed that while subtracting the mean of the log magnitude spectrum improved ASR performance, the improvement was not sufficient, especially in the presence of significant late reverberation. A reverberation compensation method for speaker recognition using spectral subtraction [29], in which late reverberation is treated as additive noise, was proposed in [4], while a method based on multistep linear prediction (MSLP) was proposed in [5,17] for both single and multiple microphones. This method first estimates late reverberation using long-term MSLP and then suppresses this with the subsequent spectral subtraction. Wang et al. proposed a distant-talking speech recognition method based on generalized spectral subtraction (SS) [30] employing the multichannel least mean squares (MCLMS) algorithm [13,31,32]. The authors further extended their method to distant-talking speaker recognition and proposed an efficient computational method for combining the likelihoods of dereverberant speech using multiple compensation parameter sets [23]. The drawback of the above approaches is that the estimation of late reverberation is not very accurate, and thus, adequate improvement cannot be achieved.

To construct a more robust representation of each cepstral feature distribution, a feature warping method was proposed [4,33]. Such methods warp the distribution of a cepstral feature stream to a standardized distribution over a specified time interval. In addition, a feature transformation approach was presented for robust distant-talking speaker recognition [34]. The transformation is applied to distorted features before mapping them to a normal distribution and aims to decorrelate the feature vectors making them more amendable to the diagonal covariance Gaussian mixture model (GMM).

Neural network-based approaches have been proposed for feature mapping and dereverberation for speech/speaker recognition [35,36] because of their flexible representations. Bottleneck features extracted by a multilayer perceptron (MLP) can be used for nonlinear feature transformation and dimensionality reduction [35]. The MLP is trained by a backpropagation algorithm from random initial parameters. Then, the bottleneck features are extracted by dimensionality reduction of several frames of cepstral coefficients. The combination of bottleneck features and cepstral coefficients is better than the conventional mel-frequency cepstral coefficients (MFCCs). However, deep networks of MLPs with many hidden layers have a high computational cost and cannot learn in layers further away from the top layer. Nugraha et al. proposed a neural network-based method to map a reverberant feature in a log-melspectral domain to its corresponding anechoic feature [36]. The results show that cascading neural network-based dereverberation significantly improves speaker recognition compared with other dereverberation approaches. Many studies have shown that cepstral features such as MFCCs are very efficient for speaker recognition; however, extending this method directly to cepstral domain dereverberation is very difficult.

Recently, deep neural network (DNN)-based approaches have been successful in many speech and image processing fields [37-40]. Deep belief networks, which employ an unsupervised pre-training method using a restricted Boltzmann machine (RBM) [39,41], have also been proposed to train better initial values of deep networks [37]. DNNs with pre-training achieve better performance than, for example, conventional MLPs without pre-training on ASR [39,40] and large vocabulary business search tasks [38]. Denoising autoencoders (DAEs) have been shown to be effective in many noise reduction applications because higher level representations and increased flexibility of the feature mapping function can be learned [42,43]. Ishii et al. applied a DAE to spectral domain dereverberation [44] and found that the word accuracy of large vocabulary continuous speech recognition improved from 61.4% to 65.2% for the JNAS (speech corpus for large vocabulary continuous speech recognition research) database [45]. However, the suppressed spectral domain feature

needs to be converted to a cepstral domain feature, and the subsequent performance improvement is not sufficient.

Few studies have focused on a DNN-based approach for distant-talking speaker recognition. By removing reverberation, we can expect to improve the speech/speaker recognition performance. However, very little research has focused on the differences between speech and speaker recognition in a distant-talking environment. For speech recognition, it is necessary to maximize the inter-phoneme variation while minimizing the intra-phoneme variation in the feature space. For speaker recognition, on the other hand, the focus is on speaker variation instead of phoneme variation. These characteristics mean some methods that are effective in speech recognition may not be as effective for speaker recognition, especially in a hands-free environment [46]. Therefore, the effect of DNN-based feature mapping and dereverberation on distant-talking speaker recognition is still unknown.

In our preliminary experiment, we found that DNN-based cepstral domain feature mapping is efficient for distant-talking speaker recognition [47]. In this paper, we present DNN-based bottleneck feature mapping, DAE-based cepstral domain dereverberation, and a combination of the two for distant-talking speaker recognition. For the DNN-based bottleneck feature (BF-DNN), we noted that DNNs can transform the reverberant speech feature to a new feature space with greater discriminative classification ability for distant-talking speaker recognition. In addition, by using multiple contexts (frames) for input data, the bottleneck features can reduce the influence of reverberation over several frames.

For neural network-based dereverberation, previous studies have shown that the spectral domain feature is efficient for the ASR task [44]. Noting that many speaker recognition systems adopt a cepstral domain feature as the direct input, it is meaningful to discover the performance of the cepstral domain DAE-based dereverberation method. Cepstral domain DAE-based dereverberation transforms the cepstrum of reverberant speech to that of clean speech. Moreover, the dimensions of the spectral domain-based features are greater than those of the cepstral domain-based ones. This introduces greater difficulties in learning a DAE with a deep architecture. Thus, it is expected that DAE-based cepstral domain dereverberation would be more efficient than DAE-based spectral domain dereverberation for speaker identification under distant-talking environments.

The DNN-based bottleneck feature is a method for extracting discriminant features while DAE-based dereverberation is a method for suppressing reverberation. Thus, they have a strong complementary nature, and a combination of the two methods should be very efficient in distant-talking speaker identification. Therefore, the likelihood of the bottleneck features extracted from the DNN and that of cepstral domain DAE-based dereverberation are combined linearly. A block diagram of the complete system is shown in Figure 1. In the training stage, DAE and BF-DNN models for feature transformation and speaker models with transformed features are trained. In the test stage, first, MFCCs extracted from the reverberant speech are input to the DAE and BF-DNN models for feature transformation. Then, the transformed features and speaker models are used to calculate the likelihood of each speaker. Finally, the likelihoods of DAE-based and BF-DNN-based features are combined and the target speaker is determined.

We also analyzed the optimal neural network architecture and parameters of the DNN-based bottleneck feature and DAE-based dereverberation for distant-talking speaker identification.

The remainder of this paper is organized as follows: Section 2 presents some basic theory for constructing and training DNNs, while an outline of the DNN-based bottleneck feature and DAE-based dereverberation method is given in Section 3. Section 4 discusses the development and evaluation of an experiment for distant-talking speaker recognition in reverberant environments. Finally, Section 5 summarizes the paper.

2 Overview of restricted Boltzmann machine

In speech recognition, DNN has been successfully used for modeling the posterior probability of state. In this work, for non-linear feature transformation, we used DNN, which can suppress the reverberation and transform the original feature to a discriminative feature for reverberant speech. A basic training strategy involved multiple phases. First, pre-training of the DNN was accomplished by training an unsupervised RBM and stacking them in a deep belief network (DBN). Second, optimization with back-propagating, referred to as fine-tuning, discriminatively trains the DNN using supervised signals. Meanwhile, in the pre-training phase of the DAE task, the encoder network was also trained layer by layer as a stack on RBM. In this section, we briefly introduced the RBM [39,41].

2.1 Restricted Boltzmann machine

The RBM is a bipartite graph as shown in Figure 2.

It has both visible and hidden layers in which visible units representing observations are connected to hidden units that learn to represent features using weighted connections. An RBM is restricted in that there are no visible-visible or hidden-hidden connections. Different types of RBMs are used for binary and real-valued input. Bernoulli-Bernoulli RBMs are used to convert binary stochastic variables to binary stochastic variables, while Gaussian-Bernoulli RBMs are used to convert

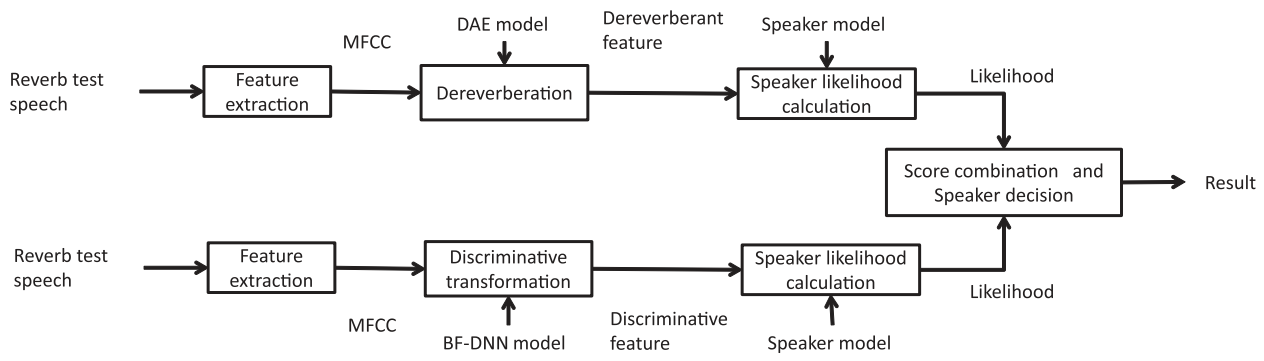

Figure 1 Block diagram of the proposed speaker recognition system.

real-valued stochastic variables to binary stochastic variables.

In a Bernoulli-Bernoulli RBM, the weights on the connections and the biases of the individual units define a probability distribution over the joint states of the visible and hidden units via an energy function. The energy of the joint configuration is given by

$$E(\mathbf{v},\mathbf{h}|\theta) = -\sum_{i=1}^{\mathcal{V}}\sum_{j=1}^{\mathcal{H}} w_{ij}v_i h_j - \sum_{i=1}^{\mathcal{V}} a_i v_i - \sum_{j=1}^{\mathcal{H}} b_j h_j, \quad (1)$$

where $\theta = (\mathbf{w},\mathbf{a},\mathbf{b})$ and w_{ij} represents the symmetric interaction term between visible unit i and hidden unit j with a_i and b_j their respective bias terms. $\mathcal{V}$ and $\mathcal{H}$ denote the numbers of visible and hidden units, respectively.

The maximum likelihood estimation of an RBM is to maximize the log likelihood $\log p(\mathbf{v}|\theta)$ of parameter θ. Therefore, the weight update equation is given by

$$\Delta w_{ij} = \epsilon(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{model}}), \quad (2)$$

where ϵ is the learning rate, $\langle\cdot\rangle_{\text{data}}$ is the expectation that v_i and h_j are on together in the training set, while $\langle\cdot\rangle_{\text{model}}$ is the same expectation calculated from the model. Because computing $\langle v_i h_j\rangle$ is expensive, we use a contrastive divergence approximation to compute the gradient. It is possible to compute $\langle v_i h_j\rangle$ by applying Gibbs sampling.

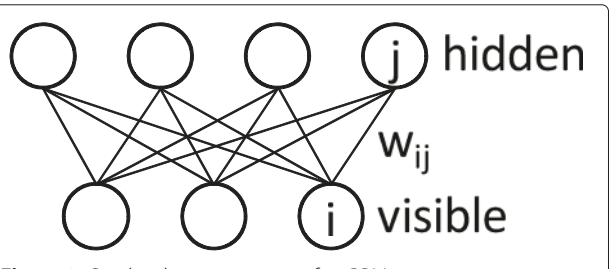

Figure 2 Graphical representation of an RBM.

2.2 DNN structure and training

DBNs are configured hierarchically by connecting pre-trained RBMs. The top layer of a DBN is a softmax layer, with the softmax operation given as

$$p(l|\mathbf{h}) = \frac{\exp(b_l + \sum_i h_i w_{il})}{\sum_m \exp(b_m + \sum_i h_i w_{im})}, \quad (3)$$

where b_l is the bias of the label and w_{il} is the weight of hidden unit i in the top layer to label l.

After configuring the DBN using RBMs, it is discriminatively trained using the backpropagation algorithm [48] to maximize the log probability of the class labels. In general, after discriminative training, a DBN is called a DNN.

In particular, we used the algorithm from [37] to train a DNN. In the pre-training phase, we first initialized the RBMs with random values. We then subdivided all training datasets into mini-batches, with 128 data vectors for unsupervised pre-training. Each hidden layer was pre-trained for 50 passes. The weight was updated after each mini-batch. For the DNN training phase, also referred to as the fine-tuning phase, we used the method of the conjugate gradient algorithm. We repeated the fine-tuning for 100 epochs updating the entire training set. The learning rate for the weights was 0.03 and for biases was 0.1.

3 DNN-based bottleneck feature and DAE-based dereverberation

3.1 Bottleneck features extracted from a DNN

Bottleneck features were generated from an MLP [35] in which one of the internal layers has a small number of hidden units relative to the size of the other layers. The multilayer network to obtain the bottleneck features is shown in Figure 3. In this example, the number of hidden layers (including the bottleneck layer) is set to 5. The number of hidden units in the innermost layer is smaller than that in the other layers. We call this the bottleneck layer.

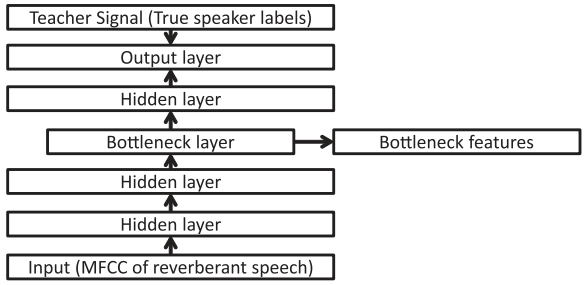

Figure 3 Flowchart of bottleneck feature extraction. In the training stage for a bottleneck DNN, the input of the DNN was a frame, or multi frames, of reverberant speech. The teacher signal was the true speaker labels. For a trained DNN, giving a frame or multiframes of reverberant speech, we took the bottleneck layer as the output feature.

In our work, both MLPs without pre-training and DNNs with pre-training were used as multilayer networks. In the pre-training step, we trained each layer of the RBM to construct a DBN using the common DBN training. With the pre-training step, the DBN achieved better initial values of the neural network. This structured bottleneck layer could be treated as a nonlinear mapping of input features. In addition, it was possible to enhance the identification ability of bottleneck features by discriminative training, which was expected to mitigate the influence of reverberation on speaker identification.

We used the speaker labels as the teacher signal. DNN's can be trained by backpropagating derivatives of a cost function that measures the cross entropy between the target outputs and the actual outputs produced for each training case.

The initial value of the MLP was generated randomly in the range −0.5 to 0.5, while the initial value of the DBN was determined by unsupervised pre-training. After initialization, supervised discriminative training was performed for both the MLP without pre-training and DBN with pre-training. Finally, the bottleneck features extracted from the bottleneck layer of the DNN were used to train the speaker model.

3.2 Denoising autoencoder for cepstral domain dereverberation

An autoencoder is a type of artificial neural network whose output is a reconstruction of the input and which is often used for dimensionality reduction.

The autoencoder training phase aims to find a value for the parameter vector, which minimizes the value between the input and teacher signals. This minimization is usually carried out by minimizing the cross entropy using conjugate gradients. Because it was difficult to directly optimize weights in a deep autoencoder with many layers, an initialization step called pre-training was conducted.

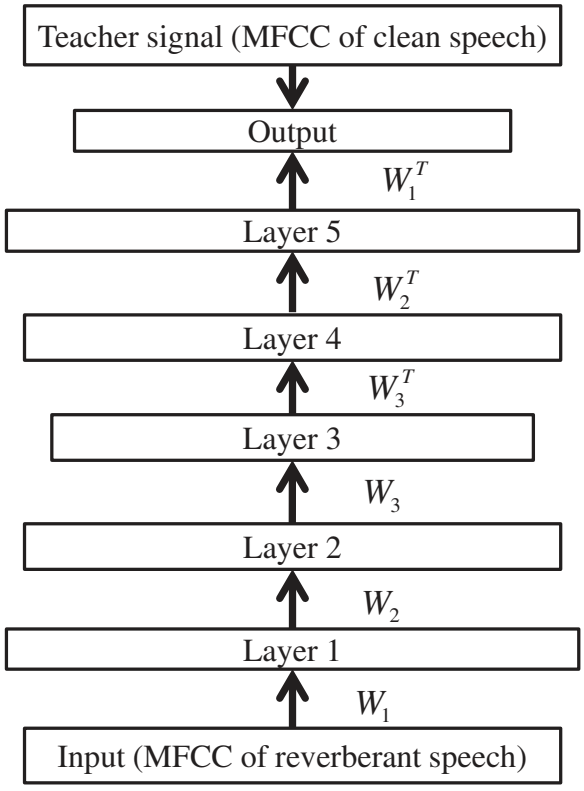

Figure 4 Topology of denoising autoencoder for cepstral domain dereverberation. In the training phase of the DAE, the cepstrum of reverberant speech was set as the input, and the cepstrum of the corresponding clean speech was set as the teacher signal. After training, the DAE network was able to generate the output of reverberation suppressed features.

DAEs share the same structure as autoencoders, but the input data are a noisy version of the output data. Autoencoders use feature mapping to convert noisy input data into clean output and, thus, have been used for noise removal in the field of image processing [42,49]. Ishii et al. applied a DAE to spectral domain dereverberation [44]. However, the suppressed spectral domain feature needs to be converted to a cepstral domain feature, and this improvement in performance was inadequate. In this

Table 1 Dataset descriptions

Data type	Usage	Data set
Training data	To train the DNN and GMM	100 (speakers) × 5 (utterances) × 3 (environments)
Development data	To determine the settings of the DNN (layers, batches, etc.) in the experimental step	Same as above
Test data	To test the speakers in this dataset in the evaluation step	100 (speakers) × 20 (utterances) × 5 (environments)

Table 2 Details of recording conditions for impulse response measurement

Array number	Room	Array type	RT60 (s)
(a) CENSREC-4 database for training			
1	Japanese style room	Linear	0.40
2	Japanese style bath	Linear	0.60
3	Elevator hall	Linear	0.75
(b) RWCP database for testing			
4	Echo room (cylinder)	Circle	0.38
5	Tatami-floored room (S)	Circle	0.47
6	Tatami-floored room (L)	Circle	0.60
7	Conference room	Circle	0.78
8	Echo room (panel)	Linear	1.30

RT60 (second), reverberation time in room. S, small. L, large.

paper, we applied a DAE for cepstral domain dereverberation because there were many speaker recognition systems that adopted cepstral domain features as their direct input. It is meaningful to evaluate the performance with cepstral domain-based DAE features of speaker recognition. Given a pair of speech samples, that is, clean speech and the corresponding reverberant speech, the DAE learns the nonlinear conversion function that converts reverberant speech features into clean speech. In general, reverberation is dependent on both the current and several previous observation frames. In addition to the vector of the current frame, vectors of past frames were concatenated to form input.

For cepstral feature X_i of the observed reverberant speech of the $i-th$ frame, cepstral features of $N-1$ frames before the current frame are concatenated with those of the current frame to form a cepstral vector of N frames. Output O_i of the nonlinear transformer based on the DAE is given by

$$O_i = f_L(\ldots f_l(\ldots f_2(f_1(X_i, X_{i-1}, \ldots, X_{i-N}))), \quad (4)$$

where f_l is the nonlinear transformation function in layer l, and N is the number of frames to be used as input features.

The topology of the cepstral domain DAE for dereverberation is shown in Figure 4. In this example, the number of hidden layers was set to five. In Figure 4, $W_i (i = 1, 2, 3)$ denotes the weighting of the different layers and W_i^T shows the transposition of W_i [a]. That is, W_1, W_2, and W_3 were the encoder matrices and W_1^T, W_2^T, and W_3^T were the decoder matrices, respectively. To train a DAE, we used DBNs [50] for pre-training because they can obtain accurate initial values of the deep-layer neural networks. To obtain a pre-trained RBM, we trained the second hidden layer using a Bernoulli-Bernoulli RBM and the third hidden layer using a Gaussian-Bernoulli RBM. DBNs are hierarchically configured by connecting these pre-trained RBMs. Here W_1, W_2, and W_3 are learned automatically, while W_1^T, W_2^T, and W_3^T are generated from W_1, W_2, and W_3, respectively.

After pre-training, a backpropagation algorithm was applied to adjust the parameters of autoencoder. Backpropagation algorithm modified the weights of autoencoder to reduce the cross entropy error between the teacher signal and the output value when a pair of signals is given (an input signal and an ideal teacher signal pairs.). In this paper, the input signal is the cepstral feature of reverberant speech and the ideal teacher signal is the cepstral feature of clean speech. The conjugate gradient algorithm was used to adjust the relative weightings of the units to minimize the cross entropy error for each training case [37].

(a) CENSREC-4 (b) RWCP

Figure 5 Graphic illustration of microphone array. **(a)** CENSREC-4 **(b)** RWCP.

Table 3 Channel numbers corresponding to Figure 5 used for dereverberation

	Linear array	Circular array
CENSREC-4	1, 3, 5, 7	—
RWCP	17, 21, 25, 29	1, 5, 9, 13

3.3 Combination of DNN-based bottleneck feature and DAE-based dereverberation

We used a GMM as our speaker model owing to its convenience and effectiveness in conventional speaker recognition. In this paper, our methods were combined by GMM likelihood. The likelihood of a DNN-based bottleneck feature-based GMM likelihood was linearly coupled with that of the DAE-based one to produce a new score L^n_{comb} given by

$$L^n_{\text{comb}} = (1-\alpha)L^n_{\text{BF}} + \alpha L^n_{\text{DAE}}, n = 1, 2, \ldots, N, \quad (5)$$

where L^n_{BF} and L^n_{DAE} are the likelihoods produced by the n-th bottleneck feature-based model and DAE-based model, respectively. N was the number of speakers registered and α denoted the weighting coefficients. The speaker with the maximum likelihood was selected as the target speaker.

4 Experiments

Our proposed method was evaluated on both simulated and actual data. Settings for the simulated data and speaker identification experiment are discussed in Section 4.1, while experimental results are presented in Sections 4.2.1 to 4.2.3. Section 4.2.1 describes the development experiment, while Section 4.2.2 evaluates our proposed method on simulated data. Section 4.2.3 investigates the effect of different training data. Regarding the experiment on actual data, details of the training data (comprising artificially created reverberant speech), the actual evaluation data, and evaluation experiment are described in Section 4.2.4.

Table 4 Reverberation methods

	Reverberation methods	
	Conventional methods	
1	CMN	MFCC with CMN
2	MCLMS-SS	Multichannel least mean squares with spectral subtraction
3	MSLP-SS	Multistep linear prediction with spectral subtraction
4	BF-MLP	Bottleneck feature extracted from multilayer perceptron
	DNN-based feature transformation methods	
5	BF-DNN	Bottleneck feature extracted from deep neural network
6	DAE	Denoising autoencoder-based cepstral-domain dereverberation
7	DAE + BF-DNN	Combination of DAE and BF-DNN

Table 5 Conditions for speaker recognition

	Values
Sampling frequency	16 kHz
Frame length	25 ms
Frame shift	10 ms
Feature space	25 dimensions with CMN (12 MFCCs + Δ + Δpower)
Acoustic model	GMMs with 128 diagonal covariance matrices

4.1 Experimental setup

We used clean speech convoluted with various impulse responses to generate simulated data for the dereverberation experiment. For the simulated data, eight multichannel impulse responses were selected from the Real World Computing Partnership (RWCP) sound scene database [51] and the CENSREC-4 database [52]. These were convoluted with clean speech to create artificial reverberant speech. A large-scale database, the Japanese Newspaper Article Sentence (JNAS) [45] corpus, was used as the source for clean speech. Table 1 describes the development, training, and test datasets. Since the training and development datasets are the same, we refer to both as the training dataset. Utterances from 100 speakers (50 male and 50 female) were used for development and to train parameters for the DAE, BF-DNN, and GMMs. For each speaker, we used three types of artificial impulses (CENSREC-4) convoluted into five different sentences unless there was a special expression. Thus, in total, 1,500 sentences (15 sentences per speaker × 100 speakers) were used to train the DAE, BF-DNN, and GMMs. Each speaker provided 20 utterances for the test data. The average duration of training and test utterances was about 3.9 and 5.6 s, respectively.

Table 2 lists the impulse responses for the training and test sets. The impulse responses were collected by microphone arrays, as illustrated in Figure 5. The channel numbers corresponding to Figure 5 used for

Table 6 Initial parameters of DNN

	Values
Number of layers	5
Number of units in each layer	1,024
Context size	9

Table 7 Learning parameters of DNN

	Value
Batch size	128
Learning rate of Bernoulli-Bernoulli RBM	0.02
Learning rate of Gaussian-Bernoulli RBM	0.002
Weight decay	0.0002
Number of iterations in pre-training	50
number of iterations in fine-tuning	100

dereverberation are shown in Table 3. For the RWCP database, a four-channel circular or linear microphone array was taken from a circular + linear microphone array (30 channels). The circular array had a diameter of 30 cm. The microphones in the linear microphone array were located at 2.83-cm intervals. Impulse responses were measured at several positions 2 m from the microphone array. For the CENSREC-4 database, four-channel microphones were taken from a linear microphone array (seven channels in total), with the microphones located at 2.125-cm intervals. Impulse responses were measured at several positions 0.5 m from the microphone array.

In this study, we compared seven dereverberation methods, briefly described in Table 4.

For each method, we performed delay-and-sum beamforming. For *method 1*, only CMN with beamforming was used to reduce the reverberation (denoted as 'CMN'). For comparison, MCLMS-SS- [32] and MSLP-SS [17]-based dereverberation was performed in *method 2* and *method 3*, respectively. The MCLMS-SS and MSLP-SS methods both treated late reverberation as additional noise and used the spectral subtraction method to suppress it. We also performed bottleneck feature extraction without pre-training, denoted as 'BF-MLP' (*method 4*) [35]. *Method 5* (denoted as 'BF-DNN'),*method 6* (denoted as 'DAE'), and *method 7* (denoted as 'DAE + BF-DNN') represent methods introduced in this paper. For all methods, dereverberant speaker models were trained using artificial reverberant speech with three types of CENSREC-4 impulse responses (see Table 2a) and suppressed by the corresponding dereverberant method. The features of dereverberant speech were used to train the dereverberant speaker models.

Table 5 lists the conditions for speaker identification. We used 25-dimensional MFCCs and GMMs [53,54] with 128 mixtures. The MFCC features were normalized with the mean of the entire training data. GMMs were trained using three kinds of reverberant speech corresponding to three kinds of impulse responses. The conditions for the MCLMS and MSLP-based methods were the same as those in [55] and [5], respectively. In the MCLMS and MSLP methods, the spectral floor parameter was set to 0.15, while the noise overestimation factor and exponent parameter were set to 0.5.

Bottleneck and DAE features for distant-talking speaker identification were extracted from the MFCC features. Since the details of the parameters of a DNN are determined by the training data, this is discussed in the next section.

Table 8 Recognition rates of DNN with varying numbers of units in each layer for training data (%)

Method	Number of units	CENSREC-4 database			Ave.
	in each layer	0.40	0.60	0.75	
BF-DNN	512	84.90	78.95	83.15	82.33
BF-DNN	1,024	88.40	83.90	88.10	86.80
BF-DNN	2048	87.45	83.25	87.05	85.92

Table 9 Recognition rates of BF-DNN with and without pre-training for training data (%)

Method	Pre-training	CENSREC-4 database			Ave.
		0.40	0.60	0.75	
BF-DNN	With	88.40	83.90	88.10	86.80
BF-MLP	Without	84.85	78.15	83.50	82.17

4.2 Experimental results

4.2.1 Results of simulated development experiment

Since the number of DNN layers and units in each layer need to be set before DNN learning, we used development experiments to determine the optimal parameter settings for this approach. The initial numbers of DNN structures were set according to Table 6, while the parameters were set according to Table 7, based on the settings in [39]. Five utterances taken from each of 50 male and 50 female speakers were used as training data (Table 1).

The structure of a DNN is determined by: 1) the units in each layer; 2) presence or absence of pre-training; and 3) the number of layers. These parameters for the bottleneck feature DNN and DAE were determined empirically.

Table 10 Recognition rates of BF-DNN with a varying number of layers for training data (%)

Method	Number of layers	CENSREC-4 database			Ave.
		0.40	0.60	0.75	
BF-DNN	3	56.60	48.65	55.45	53.57
BF-DNN	5	88.40	83.90	88.10	86.80
BF-DNN	7	90.40	86.40	89.20	88.67
BF-DNN	9	91.80	88.35	91.15	90.43

Table 11 Recognition rates of DAE with different sized contexts for the training data (%)

Method	Context size	CENSREC-4 database			Ave.
		0.40	0.60	0.75	
DAE	9 (c1+p8)	94.30	90.85	93.80	92.98
DAE	9 (c1+p4+l4)	93.40	89.60	93.15	92.05
DAE	9 (c1+l8)	93.50	89.65	92.75	91.07

Determining the DNN for bottleneck features (BF-DNN)

First, we determined the units in each layer. Table 8 shows the speaker recognition rates for the bottleneck feature DNN (denoted as BF-DNN in the table) with different unit settings in each layer. Initially, we set the number of layers to five. In theory, more units in each layer achieve better performance in recognition tasks. Conversely, too large a number of units may lead to over-learning, which causes diminished performance. Moreover, in the bottleneck layer, we need to compress the units to 25 dimensions. Thus, we chose 1,024 as the optimal number of units for a BF-DNN in the evaluation experiment.

Next, we investigated whether unsupervised pre-training is necessary for a BF-DNN. With pre-training, the BF-DNN achieves better performance (Table 9). The reason for this is that the multiple layers of the neural network present a much better starting point for a discriminative phase and converge faster [37]. We refer to this phase as 'fine turning'. Thus, pre-training was applied in the evaluation step.

Third, we considered how many layers would be appropriate for the BF-DNN. Table 10 shows the effect of the number of layers in the BF-DNN. We can see that more layers achieve better performance. Because the value of the teacher signal is significantly different from that of the input signal, the system needs more layers to transform the MFCC of reverberant speech to a teacher signal (true speaker label). Moreover, it has been shown that with fewer layers, the recognition performance of a system is extremely poor. We used nine layers for the BF-DNN in the evaluation experiment because more layers would have increased the time needed to train the BF-DNN, while resulting in only a relatively modest improvement in performance.

Table 12 Recognition rates of DAE with varying numbers of layers for training data (%)

Method	Number of layers	CENSREC-4 database			Ave.
		0.40	0.60	0.75	
DAE	1	95.35	90.65	95.20	93.73
DAE	3	95.40	91.50	94.70	93.87
DAE	5	94.30	90.85	93.80	92.98
DAE	7	89.65	84.40	87.60	87.22
DAE	9	87.85	82.75	87.50	86.03

Table 13 Optimal parameters for BF-DNN

	Values
Number of layers	9
Number of units in each layer	1,024
Context size	9 (p4 + c1 + l4)

Determining the DNN for the denoising autoencoder

In DAE learning, because of the duration of the reverberation, we cannot fully represent reverberation in a single frame. Thus, in the input layer, not only the current frame but also its neighboring frames are needed. Here, we need to determine another parameter that controls the input vector size, namely, context-size. We compared three kinds of contexts with a context size of nine: 1) left context (the current frame (c1 for short) + the previous eight frames (p8)); 2) left and right contexts (p4 + c1 + next 4 frames (l4)); and 3) right context (c1 + p8) (see Table 11). The results show that the best performance was obtained with a context size of nine (c1+ p8) in Table 11 because in a reverberant environment, the current frame could be affected by the previous frames. Thus, we used the setting of c1+ p8 in the subsequent step.

Units in the DAE refer to a setting in a BF-DNN. Pre-training is needed in the DAE for the same reason as in the BF-DNN.

The number of layers in the DAE must also be predetermined. Table 12 shows the effect of the number of layers on the DAE. Contrary to the BF-DNN, fewer layers result in better performance. This can be explained by the complex structure of DNNs: too many layers cause an increase in the transformation magnitude of fine-tuning convergence, with the output being overlearned. Contrary to the BF-DNN, the values of the input MFCC of reverberant speech and teacher signal MFCC of clean speech are similar. An appropriate number of layers is sufficient for this task. Based on the experimental results, we used three layers for the DAE in the evaluation experiment.

Table 14 Optimal parameters for denoising autoencoder

	Values
Number of layers	3
Number of units in each layer	1,024
Context size	9 (c1+p8)

Table 15 Distant-talking speaker identification rates for evaluation data (%)

Method	RT60 of test data (s) (RWCP data)					Ave.
	0.38	0.47	0.60	0.78	1.30	
(a) Conventional methods						
CMN	79.70	76.05	75.55	74.40	75.75	76.29
MCLMS-SS	82.25	79.70	78.75	78.05	81.30	80.01
MSLP-SS	82.85	78.60	78.50	78.00	75.70	78.73
BF-MLP	72.35	69.30	64.05	64.90	63.25	66.70
(b) DNN-based feature transformation methods						
BF-DNN	87.90	84.95	82.45	84.00	82.15	84.29
DAE	92.10	89.70	87.60	89.45	88.10	89.39
DAE + BF-DNN	94.20	92.20	90.65	91.95	90.70	91.94

Determining the parameters for the combination of the two systems

Because the BF-DNN system tries to find discriminative features while the DAE system aims to create a transformation that can transform reverberant features into clean features, these two systems are complementary in nature and both perform well. Thus, we considered that a combination of likelihoods of these two systems could achieve better performance using Equation 5. To determine the linear combination parameter α, we varied α from 0.1 to 0.9 in steps of 0.1 and computed the recognition rate in each step. The maximum recognition rate was obtained for $\alpha = 0.4$. Thus, this value was used in the evaluation experiment.

4.2.2 Experimental results of simulated evaluation data

The optional parameters determined in the development step are summarized in Tables 13 and 14. Table 15 compares the results of distant-talking speaker identification using conventional and the proposed methods. The results show that both BF-DNN and DAE performed better than the conventional methods in all five different reverberant environments. The speaker recognition rates based on CMN for distant-talking conditions are very low because late reverberation cannot be suppressed. Conventional late reverberation suppression methods such as MCLMS-SS and MSLP-SS do not achieve sufficient improvement either. Both of the DNN-based feature mapping methods outperform the conventional dereverberation methods. Our nonlinear transformation-based approaches have a more flexible representation ability, which is more suited to distant-talking speech with a complex distribution.

The reason for the improvement in BF-DNN learning is illustrated directly in Figure 6. For the BF-DNN, we performed linear discriminant analysis to reduce the dimensions of the utterances of 20 speakers from 25 to 2 and showed them in two directions of coordinating axes. The distribution of speaker's features is clearly distinguished here. The BF-DNN changes the features to a space that is easily distinguished. We also applied the proposed method with and without pre-training (bottleneck feature MLP in Table 15). The pre-trained method achieved better performance. The unsupervised pre-training step enhanced the distinguishing characteristics (in our experiment, these are the dereverberation characteristics) to obtain good initial parameters for the neural network. The supervised training step then leads these distinguishing characteristics in the right direction, hence the need for a pre-training step.

As for the DAE, the improvement in recognition rate can be explained by the fact that the DAE always learns

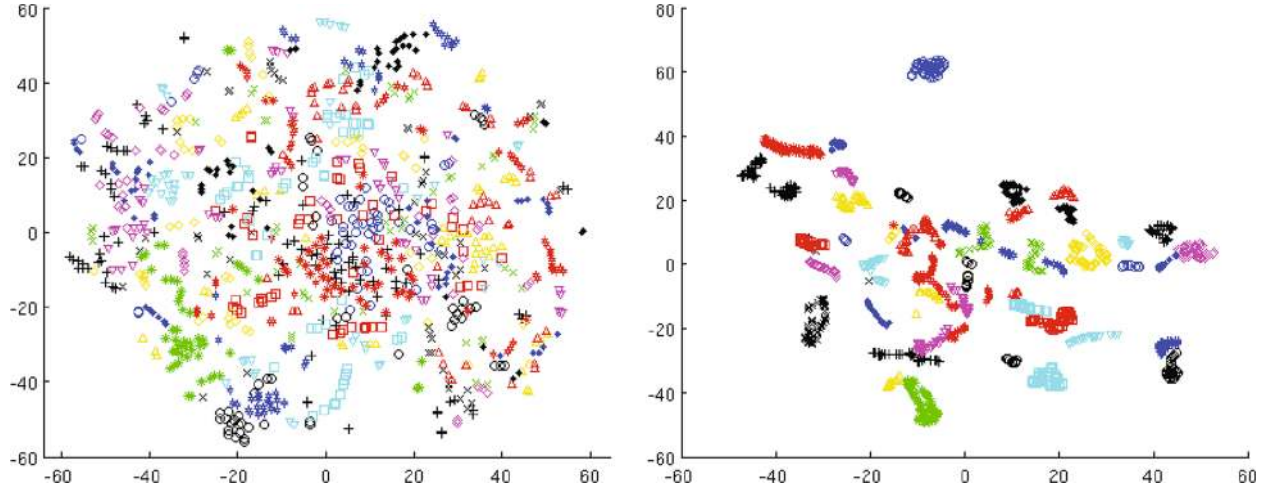

Figure 6 Feature space before and after the BF-DNN. We used PCA to reduce the speaker features to two dimensions. The lift graph shows the locations of the different speakers using different colors in a two-dimensional plane. The boundary of each speaker is not clear. The right graph shows the features after the BF-DNN in two dimensions. Here, the boundaries are clearer than those on the left graph.

Table 16 Three different training sets used to train the DNN and GMM

Training set number	DAE and BF-DNN training data	GMM training data
1	5 (utterances) × 100 (speakers) × 3 (environments)	5 (utterances) ×100 (speakers) × 3 (environments)
2	Same as training set 1	10 (utterances) ×100 (speakers) × 3 (environments)
3	10 (utterances) ×100 (speakers) × 3 (environments)	Same as training set 2

a vector field toward the higher probability regions and minimizes the variational lower bound on a generative model [42].

The DAE retains speaker characteristics and suppresses the reverberation by nonlinear feature mapping. The BF-DNN classifies speaker characteristics in the right direction. Therefore, BF-DNN-based discriminant features and DAE-based dereverberation have a strong complementary nature. A linear combination of the likelihoods of the DAE and BF-DNN was also evaluated. We used Equation 5 to obtain the combination. The weights of the DAE and BF-DNN likelihoods were 0.6 and 0.4, respectively, corresponding to the settings in the development experiment. The combination method performed better than all the individual methods. The average reduction in relative error rate was 66.0%, 59.7%, and 62.1% for CMN, MSLP-SS, and MCLMS-SS methods.

4.2.3 *Investigation of the effect of varying sizes of training data*

We also investigated how the result changes with a varying amount of training data. In this experiment, we doubled the training data for each speaker and compared the recognition results with the different training sets. Details of the training sets used in this section are given in Table 16. For all training sets, the test set was the same (20 sentences per speaker × 5 environments).

Variations in the results are shown in Figure 7. Recognition performance improves with more training data. Using twice as much GMM training data and retaining the same training data to train the DNN-based feature transformation model (experiment 2), the relative error rates of all methods were reduced by more than 40%. When doubling the size of the training data for both GMMs and DNNs, the recognition results of DNN-based feature transformation approaches are further improved. The DNN-based method outperformed both MCLMS and MSLP-based dereverberation under all conditions.

4.2.4 *Experimental results of actual environmental data*

We also used reverberant speech from an actual environment in our experiment. The recording setting was the same as in our previous work [55]. The speech was collected in a meeting room with dimensions 7.7 m × 3.3m × 2.5m ($D \times W \times H$). The utterances were collected from 20 male speakers. Each speaker uttered nine training phrases, which were recorded by an adjacent microphone. To train GMMs, the clean speech recorded by the adjacent microphone in the actual reverberant environment was convoluted with three types of impulse responses from the CENSREC-4 database to create artificial reverberant

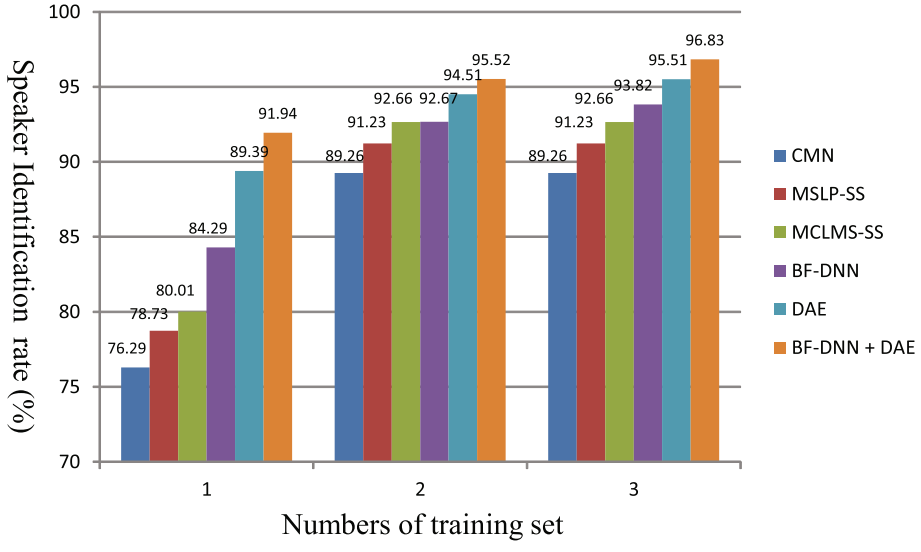

Figure 7 Average speaker identification rates using different training sets.

Table 17 Speaker identification rates in actual environment (%)

Method	Recognition rate
CMN	72.5
MCLMS-SS	87.8
MSLP-SS	83.8
BF-DNN	88.0
DAE	91.0
DAE + BF-DNN	92.5

speech. For the detailed conditions, please refer to [55]. Thus, 540 sentences (9 sentences per speaker × 3 environments × 20 speakers) were used to train the GMMs. To avoid overlearning, the 540 sentences plus the training data (1,500 sentences) shown in Table 1 were used to train BF-DNN and DAE. We retained the same neural network settings used in the experiment with simulated data (see Tables 13 and 14). For the test data, 400 utterances (20 sentences per speaker × 20 speakers) recorded by a distant four-channel microphone array were used.

The results are shown in Table 17. For the real data, a similar tendency to that found in the simulated data was observed. DAE and BF-DNN outperformed CMN, MSLP, and MCLMS. By combining the likelihoods of the DAE and BF-DNN-based features, a further improvement was achieved.

5 Conclusions

In this paper, we presented two robust distant-talking speaker identification methods based on DNNs and using bottleneck and DAE features, respectively. Bottleneck and DAE features extracted from the DNN were used to train a GMM for speaker identification. These methods achieved recognition rates of 84.29% (bottleneck DNN) and 89.39% (DAE) compared with 78.73% for the conventional MSLP and 80.01% for MCLMS in an artificial reverberant environment. Results comparing an MLP without pre-training with a DNN with pre-training show that pre-training is effective for distant-talking speaker identification. Moreover, speaker recognition performance is further improved by combining the likelihoods of the bottleneck and DAE features.

In an actual reverberant environment, BF-DNN- and DAE-based approaches also worked better than MSLP-SS and MCLMS-SS methods. The combination of DAE- and BF-DNN-based methods outperformed other methods.

Recently, Weninger et al. proposed a method for combining spectral subtraction with reverberation time estimation-based dereverberation and DAE [56]. They used reverberant and dereverberant speech to train the deep recurrent denoising autoencoder. As the DAE was trained with prior knowledge of dereverberant speech, it could learn the relationship between clean, reverberant, and dereverberant speech. This provides good motivation for our future work, that is, combining DAE with MCLMS-based dereverberation.

Endnote

[a] W_i and Wi_1^T correspond to f_L in Equation 4.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was partially supported by a research grant from the Kayamori Foundation of Informational Science Advancement.

Author details
[1]Nagaoka University of Technology, 1603-1 Kamitomioka-cho, Nagaoka, Niigata, 940-2188, Japan. [2]Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8561, Japan. [3]Tsinghua University, 1 Qinqhuayuan, Beijing 100084, China.

Received: 28 July 2014 Accepted: 16 April 2015
Published online: 12 May 2015

References

1. T Yoshioka, A Sehr, M Delcroix, K Kinoshita, R Maas, T Nakatani, W Kellermann, Making machines understand us in reverberant rooms: robustness against reverberation for automatic speech recognition. IEEE Signal Process. Mag. **29**(6), 114–126 (2012)
2. M Wu, D Wang, A two-stage algorithm for one-microphone reverberant speech enhancement. IEEE Trans. ASLP. **14**(3), 774–784 (2006)
3. M Delcroix, T Hikichi, M Miyoshi, Precise dereverberation using multi-channel linear prediction. IEEE Trans. ASLP. **15**(2), 430–440 (2007)
4. Q Jin, T Schultz, A Waibel, Far-field speaker recognition. IEEE Trans. ASLP. **15**(7), 2023–2032 (2007)
5. K Kinoshita, M Delcroix, T Nakatani, M Miyoshi, Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction. IEEE Trans. Audio Speech Lang. Process. **17**(4), 534–545 (2009)
6. H Hirsch, H Finster, A new approach for the adaptation of HMMs to reverberation and background noise. Speech Commun. **50**(3), 244–263 (2008)
7. A Sehr, R Maas, W Kellermann, Reverberation model-based decoding in the logmelspec domain for robust distant-talking speech recognition. IEEE Trans. ASLP. **18**(7), 1676–1691 (2010)
8. BD Van Veen, KM Buckley, Beamforming: a versatile approach to spatial filtering. IEEE assp mag. **5**(2), 4–24 (1988)
9. EA Habets, in *Proc. IEEE ICASSP*. Multi-channel speech dereverberation based on a statistical model of late reverberation, vol. 4, (2005), pp. 173–176
10. S Gannot, M Moonen, Subspace methods for multimicrophone speech dereverberation.EURASIP. J Appl. Signal Process. **2003**(1), 1074–1090 (2003)
11. S Subramaniam, AP Petropulu, C Wendt, Cepstrum-based deconvolution for speech dereverberation. IEEE Trans. Speech Audio Process. **4**(5), 392–396 (1996)
12. H Maganti, M Matassoni, in *Proceedings of INTERSPEECH-2010*. An auditory based modulation spectral feature for reverberant speech recognition, (2010), pp. 570–573
13. L Wang, K Odani, A Kai, Dereverberation and denoising based on generalized spectral subtraction by multi-channel LMS algorithm using a small-scale microphone array.Eurasip. J. Adv. Signal Process. **2012**(12), 1–11 (2012)
14. L Wang, N Kitaoka, S Nakagawa. Eurasip J. Appl. Signal Process. **2006**(95491), 1–11 (2006)
15. Q Jin, Y Pan, T Schultz, Far-field speaker recognition. Proc. of ICASSP-2006. **1**, 937–940 (2006)

16. SO Sadjadi, JHL Hasnen, Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions. Proc. IEEE ICASSP, 5448–5451 (2011)
17. K Kinoshita, M Delcroix, T Nakatani, M Miyoshi, in *Proceedings of IEEE ICASSP 2006*. Spectral subtraction steered by multi-step forward linear prediction for single channel speech dereverberation, (2006), pp. 817–820
18. TB Hughes, HS Kim, JH DiBiase, HF Silverman, Performance of an HMM speech recognizer using a real-time tracking microphone array as input. IEEE Trans. Speech, and Audio Process. **7**(3), 346–349 (1999)
19. S Furui, Cepstral analysis technique for automatic speaker verification. IEEE Trans. Acoust. Speech Signal Process. **29**(2), 254–272 (1981)
20. F Liu, R Stern, X Huang, A Acero, in *Proc. ARPA Speech and Nat. Language Workshop*. Efficient cepstral normalization for robust speech recognition, (1993), pp. 69–74
21. L Wang, N Kitaoka, S Nakagawa, Robust distant speaker recognition based on position dependent cepstral mean normalization. Proc. of Interspeech. **2005**, 1977–1980 (2005)
22. L Wang, N Kitaoka, S Nakagawa, in *Proc. of ICASSP*. Robust distant speech recognition by combining position-dependent CMN with conventional CMN, (2007), pp. 817–820
23. L Wang, Z Zhang, A Kai, Hands-free speaker identification based on spectral subtraction using a multi-channel least mean square approach.Proc. ICASSP. **2013**, 7224–7228 (2013)
24. TB Hughes, HS Kim, JH DiBiase, HF Silverman, Performance of an HMM speech recognizer using a real-time tracking microphone array as input. IEEE Trans. Speech Audio Process. **7**(3), 490–497 (1997)
25. ML Seltzer, B Raj, RM Stern, Likelihood-maximizing beamforming for robust hands-free speech recognition. IEEE Trans. Speech Audio Process. **12**(5), 489–498 (2004)
26. EA Habets, J Benesty, in *Hands-free Speech Commun. Microphone Arrays (HSCMA) 2011 Joint Workshop on. IEEE*. Joint dereverberation and noise reduction using a two-stage beamforming approach, (2011), pp. 191–195
27. D Gelbart, N Morgan, *Evaluating long-term spectral subtraction for reverberant ASR*. (Madonna di Campiglio, Italy, 2001)
28. D Gelbart, N Morgan Double the trouble: handling noise and reverberation in far-field automatic speech recognition. INTERSPEECH, 968–971 (2002)
29. S Boll, Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acous. Speech Signal Process. **27**(2), 113–120 (1979)
30. BL Sim, YC Tong, JS Chang, CT Tan, A parametric formulation of the generalized spectral subtraction method. IEEE Trans. Speech Audio Process. **6**(4), 328–337 (1998)
31. Y Huang, J Benesty, J Chen, Optimal step size of the adaptive multi-channel LMS algorithm for blind SIMO identification. IEEE Signal Process. Lett. **12**(3), 173–175 (2005)
32. L Wang, N Kitaoka, S Nakagawa, Distant-talking speech recognition based on spectral subtraction by multi-channel LMS algorithm. IEICE TransInf. Syst., 659–667 (2011)
33. J Pelecanos, S Sridharan, in *Proc. of Speaker Odyssey 2001 conference*. Feature warping for robust speaker verification, (2011)
34. D Zhu, B Ma, H Li, Q Huo, in *Proc. of ICASSP 2007*. A generalized feature transformation approach for channel robust speaker verification, (2007)
35. Y Konig, L Heck, M Weintraub, K Sonmez, in *Proc. of RLA2C, ESCA workshop on Speaker Recognition and its Commercial and Forensic Applications*. Nonlinear discriminant feature extraction for robust text-independent speaker recognition, (1998), pp. 72–75
36. A Nugraha, K Yamamoto, S Nakagawa, Single-channel dereverberation by feature mapping using cascade neural networks for robust distant speaker identification and speech recognition. Eurasip. J Adv. Signal Process. **2014**(13), 1–31 (2014)
37. G Hinton, R Salakhutdinov, Reducing the dimensionality of data with neural networks. Science. **313**(5786), 504–507 (2006)
38. D Yu, ML Seltzer, in *Proc. of Interspeech*. Improved bottleneck features using pretrained deep neural networks, (2011), pp. 237–240
39. G Hinton, DYu L Deng, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, B Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
40. A Mohamed, GE Dahl, G Hinton, Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Language Process. **20**, 12–22 (2012)
41. G Hinton, A practical guide to training restricted Boltzmann machines machine learning group, 2010-003 (2010)
42. P Vincent, H Larochelle, I Lajoie, Y Bengio, PA Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
43. X Lu, Y Tsao, S Matsuda, C Hori, Speech enhancement based on deep denoising autoencoder. Proc. Interspeech, 436–440 (2013)
44. T Ishii, H Komiyama, T Shinozaki, Y Horiuchi, S Kuroiwa, in *Proc. Interspeech*. Reverberant speech recognition based on denoising autoencoder, (2013), pp. 3512–3516
45. K Itou, M Yamamoto, K Takeda, T Kakezawa, T Matsuoka, T Kobayashi, K Shikano, S Itahashi, JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research. J. Acoust. Soc. Jpn. (E). **20**(3), 199–206 (1999)
46. L Wang, N Kitaoka, S Nakagawa, Robust distant speaker recognition based on position-dependent CMN by combining speaker-specific GMM with speaker-adapted HMM. Speech Commun. **49**(6), 501–513 (2007)
47. T Yamada, L Wang, A Kai, Improvement of distant-talking speaker identification using bottleneck features of DNN. Proc. Interspeech, 3661–3664 (2013)
48. D Rumelhart, G Hinton, R Williams. Learning representations by back-propagating error.Nature. **323**(6088), 533–536 (1986)
49. P Vincent, H Larochelle, Y Bengio, PA Manzagol, in *Proceedings of the 25th International Conference on Machine learning*. Extracting and composing robust features with denoising autoencoders (ACM, 2008), pp. 1096–1103
50. GE Hinton, S Osindero, YW Teh. A fast learning algorithm for deep belief nets.Neural comput. **18**(7), 1527–1554 (2006)
51. S Nakamura, K Hiyane, F Asano, T Nishiura, T Yamada, in *Proc. of LREC*. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition, (2000), pp. 965–968
52. T Nishiura, M Nakayama, Y Denda, N Kitaoka, K Yamamoto, T Yamada, S Tsuge, C Miyajima, M Fujimoto, T Takiguchi, in *Proc. of INTERSPEECH*. Evaluation framework for distant-talking speech recognition under reverberant environments, (2008), pp. 968–971
53. DA Reynolds, Speaker identification and verification using Gaussian mixture speaker models. Speech Commun. **17**(1–2), 91–108 (1995)
54. DA Reynolds, TF Quatieri, R Dunn, Speaker verification using adapted Gaussian mixture models. Dig. Signal Process. **10**(1–3), 19–41 (2000)
55. Z Zhang, L Wang, A Kai, Distant-talking speaker identification by generalized spectral subtraction-based dereverberation and its efficient computation. Eurasip J. Audio Speech Music Process. **2014**(15), 1–12 (2014)
56. F Weninger, S Watanabe, Y Tachioka, B Schuller, in *Proc. of IEEE International conference Acoustics, Speech and Signal Processing (ICASSP)*. Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition, (2014), pp. 4623–4627