INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

I-vector Transformation Using a Novel Discriminative Denoising Autoencoder for Noise-robust Speaker Recognition

Shivangi Mahto, Hitoshi Yamamoto, Takafumi Koshinaka

Data Science Research Laboratories, NEC Corporation, Japan

s-mahto@cp.jp.nec.com

Abstract

This paper proposes i-vector transformations using neural networks for achieving noise-robust speaker recognition. A novel discriminative denoising autoencoder (DDAE) is employed on i-vectors to remove additive noise effects. The DDAE is trained to denoise and classify noisy i-vectors simultaneously, making it possible to add discriminability to the denoised i-vectors. Speaker recognition experiments on the NIST SRE 2012 task shows 32% better error performance as compared to a baseline system. Also, our proposed method outperforms such conventional methods as multi-condition training and a basic denoising autoencoder.

Index Terms: speaker recognition, additive noise, i-vector transformation

1. Introduction

Speaker recognition has a wide range of applications, from security solutions in forensics to client authentication in call centers. Standard speaker recognition systems consist of two stages: speaker feature extraction followed by verification using the extracted speaker features. I-vectors have been widely used for speaker feature representation [1]. Probabilistic linear discriminant analysis (PLDA) [2, 3, 4, 5, 6, 7] is the most commonly used verifier for such i-vector based speaker recognition systems.

In real world scenarios, additive noise in speech has been a challenging problem for speaker recognition and often adversely affects performance. This problem has been addressed in many studies at various steps in speaker recognition systems.

For use in an early step of acoustic feature extraction, a deep RNN-based speech enhancement technique has been presented [8] and shown to outperform spectral-based speech enhancement approaches, such as speaker-dependent NMF [9], which assumes prior knowledge about test noise, as well as STSA-MMSE estimation [10]. However, all these approaches degrade system performance in the case of high SNR (20dB) and/or clean evaluation speech. Alternatively, a DNN-based autoencoder for speech enhancement has also been presented [11]. While it is effective in compensating for distortion caused by reverberation, in the case of additive noise it was shown to perform worse than multi-condition training of PLDA, which is a classical approach to noise-robust back-end issues [12].

In the back-end step, multi-condition training in PLDA uses a large amount of both clean and noisy data, the latter of which is obtained by adding a variety of noise to the clean data at various SNR levels. The method is effective in general, but it performs sub-optimally if there is a mismatch between training and evaluation noise. Also, with clean test speech, it introduces unwanted system noise, which results in worse performance than that of single-condition clean training.

In other studies, additive noise compensation in the i-vector space has also been introduced. One such method, based on MAP [13], assumes that additive noise in a signal space has a linear effect and a Gaussian distribution in the i-vector space, which may not always be true. It requires high computational cost and is also methodologically cumbersome to apply.

Recent applications of deep learning in speaker verification have focused on i-vector transformation-based methods for various purposes, as, for example, to extract robust features from i-vectors [14, 15], to compensate for within-speaker channel distortion [16], or to restore speaker features in short speech i-vectors [17]. Dealing in i-vector space enables us to discreetly handle unwanted speaker-related variability.

This paper proposes i-vector transformations using neural networks for achieving noise-robust speaker recognition. A denoising autoencoder (DAE) has been applied on i-vectors to compensate for additive noise. The DAE is trained to denoise noisy i-vectors, which essentially minimizes the variance introduced by noise in the i-vector space. In experiments on the NIST Speaker Recognition Evaluation (SRE) task, it has been shown to be capable of handling additive noise effectively.

A DAE does not, however, use any speaker-related information. For better classification, smaller within-speaker variance and larger between-speaker variance is desired, and we propose a novel discriminative denoising autoencoder (DDAE) which is trained to denoise and classify noisy i-vectors simultaneously, making it possible to add discriminability to the denoised i-vectors. The DDAE not only removes noise effects but also adds speaker-related information in the transformed features. Our method does not put any kind of assumption on the effect of additive noise in the i-vector space or on the distribution of noise in the i-vector space. For robust speech recognition, multi-task learning has been applied in literature [18, 19, 20]. Our work is first attempt to apply such multi-task learning concept in speaker recognition.

This paper is organized as follows: Section 2 introduces key technologies used in the baseline system of speaker recognition; Section 3 presents novel i-vector transformations using neural networks; and Section 4 demonstrates significant advantage of DDAE through experimental evaluation for speaker recognition in a NIST SRE task. In Section 5, we summarize our work and discuss future work.

2. I-vector and PLDA

Recent standard speaker recognition systems consist of two stages: speaker feature extraction and verification using the extracted speaker features. I-vectors are commonly used to represent speakers in the form of a low-dimensional vectors extracted from a speech utterance by means of factor analysis. As described in [1], factor analysis makes it possible to project a speech utterance onto a low-dimensional total vari-

http://dx.doi.org/10.21437/Interspeech.2017-731

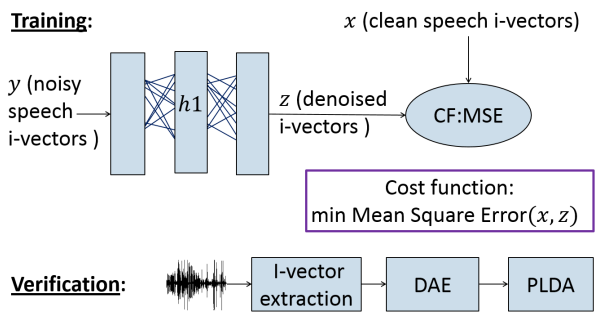

Figure 1: *Training of the DAE network and its application in the speaker verification system*

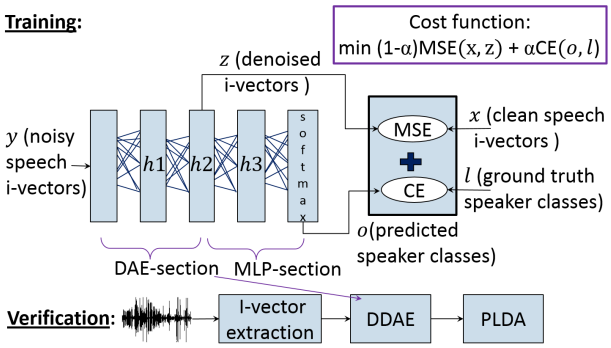

Figure 2: *Training of the Discriminative DAE network and its application in the speaker verification system*

ability space as an i-vector. Probabilistic Linear Discriminant Analysis (PLDA) [2, 3, 4] is based on a probabilistic model that is widely applied in speaker recognition as a generative linear-Gaussian model of i-vectors. A PLDA-based speaker verification system measures the similarity between two given i-vectors as a likelihood ratio.

3. I-vector transformation using neural networks

In the case of additive noise, we can express the input noisy speech signal Y as the summation of clean speech (X) and noise (N) signals:

$$Y = X + N.$$

I-vectors (i_X, i_Y) extracted from the signals X and Y can be expressed as:

$$\mathbf{i}_X = f(X),$$

$$\mathbf{i}_Y = f(Y) = f(X + N).$$

Note that the i-vector extraction process $f(\cdots)$ is a non-linear function since acoustic feature extraction and i-vector extraction from acoustic features are non-linear processes. For dealing with the non-linear effects of additive noise in i-vector space, such non-linear models as neural networks can be very effective. We have studied the application of the two neural networks described below in our speaker verification system for transforming i-vectors before sending them into PLDA for noise-robust verification.

3.1. Denoising autoencoder (DAE)

The first is the Denoising Autoencoder (DAE). An autoencoder (AE) is a neural network that learns underlying distributions for given data. A DAE is an extension of an AE, and learns non-linear mapping between corrupted and clean features [21]. We apply it to i-vector features.

Training data: It takes corrupted versions of clean data as input, and produces output which should be as close as possible to the clean data.

Objective function: Neural network updates its parameters during training to minimize mean square error (MSE) between its output and expected clean data. It can be formulated as:

$$\min MSE = \frac{1}{T}\sum_{j=1}^{T} \left\|\mathbf{x}_j - \mathbf{z}_j\right\|^2,$$

where $\mathbf{x}_j$ is the expected clean data and $\mathbf{z}_j$ is the DAE output corresponding to the jth input training sample. T is the total number of training data samples.

In our experiments, i-vectors corresponding to noisy speech (augmented clean speech with random noise) are taken as input to the DAE, and the network is trained to map them to i-vectors extracted from the corresponding clean speech. In this case, the DAE learns to minimize variance introduced by noise for each utterance.

During speaker verification, the trained DAE is applied into the system for transforming i-vectors before sending them into PLDA, in both the development and evaluation stages. It takes i-vectors as input **y** and produces transformed i-vectors as output **z**. Figure 1 shows an overview of the DAE training and its application in the speaker verification system.

3.2. Discriminative denoising autoencoder (DDAE)

As explained above, DAE is trained to minimize the variance caused by noise. It does not use any kind of speaker label information while training. Since good features should have small within-speaker variance and large between-speaker variance, there is a room for improvement in this denoising method.

To minimize within-speaker variance and maximize between-speaker variance simultaneously, we propose an extension of a denoising auto-encoder called a Discriminative Denoising Autoencoder (DDAE). Its network combines a DAE with a multi-layer perceptron (MLP) that is trained to denoise and classify noisy input data.

As shown in Figure 2, the DDAE network training can be divided into 2 sections: The first has the same structure as the DAE as it has been explained in Section 3.1. The second section consists of an MLP that takes denoised output from the first section as input and predicts the speaker label of the input data such that the cross entropy error between the predicted speaker label and the ground truth can be minimized.

The crucial part of this structure is the cost function, which jointly minimizes the mean square error for the first section and the cross entropy error for the second section. This training ensures that noisy input data will be denoised and classified correctly at the same time. The output of the first section of the DDAE network can be expected to be a denoised version of the noisy input data, as well as to have a discriminative property because of the discriminative training in the second section.

Training data: As with DAE, it takes corrupted versions of clean data as input and produces two outputs: one is a

denoised version of the input and the other is the predicted class of input, both of which should be as close as possible to the ground truth.

Objective function: The neural network updates its parameters during training to minimize the mean square error (MSE) between its denoised version of input and the expected clean data as well as the cross entropy error (CE) between the predicted class label and the corresponding ground truth. Its formulation can be expressed as:

$$\begin{aligned} &\min (1-\alpha)MSE + \alpha CE \\ &= \frac{1}{T}\sum_{j=1}^{T}\left\{(1-\alpha)\left\|\mathbf{x}_j - \mathbf{z}_j\right\|^2 + \alpha\sum_{k=1}^{K} l_j^k(\log o_j^k)\right\} \end{aligned} \quad (1)$$

where

- o_j^k is the probability predicted by DDAE that the jth training sample belongs to the kth class.
- l_j^k is the empirical (observed in the ground truth) probability that the jth training sample belongs to the kth class.
- α is the weight parameter for CE error portion.
- K is the number of classes present in the training data.

In our experiments, i-vectors corresponding to noisy input speech (augmented clean speech with random noise) are taken as input in the DDAE, and the network is trained to map them to i-vectors extracted from the corresponding clean speech and to classify them into their ground truth speaker classes.

Similar to the application of the DAE, the speaker verification system applies the first section of the trained DDAE-network to take i-vectors as input **y** and produces transformed i-vectors as output **z**, which is fed into PLDA for further processing, as shown in Figure 2.

4. Evaluation

4.1. Experimental setup

We experimentally evaluated the performance of the DAE and our proposed DDAE methods in a speaker verification task from the NIST SRE 2012 [22]. In the experiments, we used telephone speech from the male portion of Common Condition 2 (CC2, clean condition) and Common Condition 4 (CC4, additive noise condition) as trial sets. In the trial sets, the enrollment and test data were both clean for CC2, while noisy test data (additive noise) was used for CC4. Performance measures for the evaluation were the equal error rate (EER) and minimum value of the decision cost function (minDCF) for NIST SRE 2008 [23] on trials calculated with the BOSARIS toolkit [24].

For our experiments, we compared four speaker verification systems: a baseline system that does not apply any noise compensation technique; a multi-condition training-based system that uses conventional multi-condition training of PLDA [12] for the back-end step, as discussed in Section 1; a DAE-based system; and a DDAE-based system, as discussed in Section 3.

4.1.1. Baseline system

The baseline speaker verification system consisted of the i-vector and PLDA framework described in Section 2. In it, the input-speech was first converted to a time series of acoustic feature vectors, each of which consisted of 60 features (20 dimensional features consisting of 0th dimension energy features and 1-19th dimension PLP features, followed by their Δ and $\Delta\Delta$) extracted from a frame of 20 ms width for every 10 ms. An i-vector of 400 dimensions was then extracted as a speaker feature from the acoustic features using a Gaussian mixture model with 2048 mixture components as a universal background model (UBM), as well as a total variability matrix (TVM). We utilized the Kaldi speech recognition toolkit [25] to run these steps. Mean subtraction, whitening, and length normalization [26] were applied to an i-vector, as a pre-processing step before sending it to the PLDA step, and then similarity was evaluated using the simplified-PLDA model [4] with a speaker space of full 400 dimensions.

The UBM, TVM, and PLDA models were trained with development data that were different from the data in the trial list. The development data for the UBM and TVM were a combination of the SRE 2004-2010, Switchboard, and Fisher corpora. The data as a whole consisted of 30,996 utterances from 7,762 male speakers. For PLDA models, a combination of the SRE 2004-2010 and Switchboard corpora were used for training that included 17,136 utterances from 2,218 male speakers.

4.1.2. Multi-condition training-based system

The system architecture was same as the baseline system except for the PLDA training data. Unlike the baseline system, PLDA in this system used multi-condition data.

We used augmented versions of development (clean) data for PLDA as multi-condition data in our experiments. In the augmented versions, each utterance in the clean data was augmented with one of 15 noise samples taken randomly from the PRISM corpus [27] at 8, 15, and 20 dB SNR, using a publicly available tool called FaNT [28]. Experimental results showed that the multi-condition training-based system gave the best performance in CC4 trials for the multi-condition data at 8dB SNR and having 34,272 utterances from 2,218 male speakers, and we used this multi-condition data in all our experiments.

4.1.3. Neural network-based systems

We experimented on 2 neural network-based systems: a DAE-based system and a DDAE-based system. They consisted of the baseline system and a trained neural-network (either DAE or DDAE) as explained in Section 3. Note that, similar to the baseline system, PLDA training data is clean in these systems, since the output of neural-networks is assumed to be noise free. Also, in the case of DDAE-based system, whitening is disabled in the pre-processing step of PLDA. Since whitening decorrelates the input vectors, it may be possible that this mitigates the discriminative nature of i-vectors transformed by DDAE.

For the DAE, we used a single hidden layer neural network consisting of input and output layers of 400 nodes and a hidden layer of 2000 nodes. The hidden layer and output layer have, respectively, ReLU and linear activation functions.

Since the DDAE network, has 2 sections, it also has two outputs at the time of training, and these are optimized simultaneously. The first section has the same structure as that of the above-described DAE. The second section consists of an input layer of 400 nodes, a hidden layer of 2000 nodes with a ReLU activation function and an output layer with a softmax function, and it has 2218 nodes (the same as the number of speaker classes as in training data). The input layer of the second section is the output layer of the first section.

Both of the neural networks (DAE and DDAE) were trained on multi-condition data (as explained in 4.1.2).

Table 1: *Noisy test data: Equal error rate (EER, %) and minimum detection cost function (minDCF). All values were measured on Common Condition 4, SRE12 (CC4 - telephone, additive noise in test data). Our DDAE-based transformation outperforms all the other methods.*

System	EER%	minDCF
a) Baseline	4.47	0.199
b) Multi-condition training-based	3.43	0.143
c) DAE-based	3.20	0.153
d) DDAE-based	3.03	0.136

Table 2: *Clean test data: Equal error rate (EER, %) and minimum detection cost function (minDCF). All values were measured on Common Condition 2, SRE12 (CC2 - telephone, clean test data). Unlike with multi-condition training, the neural network-based transformations do not degrade performance.*

System	EER%	minDCF
a) Baseline	1.55	0.085
b) Multi-condition training-based	1.84	0.090
c) DAE-based	1.56	0.085
d) DDAE-based	1.54	0.085

We used the Keras library [29] for our neural network implementation. The structures were optimized using ADADELTA [30] with a mini-batch of 512 training samples. The number of training iterations was set to 2000. For the DDAE-based system, we tried various settings of parameter α, as shown in Figure 3, and got the best system performance for $\alpha = 0.5$.

4.2. Experimental results

Table 1 shows EERs and minDCFs in a series of speaker verification experiments for male portions of CC4 and for which trials had clean enrollment and noisy test data. The DAE-based system achieved an EER of 3.20%; 28% lower as compared to the baseline system. Also, in the case of minDCF, it performed 23% better than the baseline system which shows the capability of DAE in handling additive noise. The EER achieved with our DDAE-based system was 3.03%, as opposed to the 4.47% for the baseline system, i.e. 32% better. Our method also outperformed the multi-condition training-based system by 12% and the DAE-based system by 5%. The DDAE showed its effectiveness in minDCF as well; 32% better than the baseline, 5% better than the multi-condition training-based, and 11% better than the DAE. These results indicate that the Discriminative DAE not only denoises i-vectors but also adds speaker discriminability information, which improves speaker verification accuracy.

Table 2 shows EERs and minDCFs in a series of speaker verification experiments for male portions of CC2 and for which trials had clean enrollment and clean test data. Speaker verification using multi-condition training-based system showed an undesired degradation of 19% in EER with respect to the baseline. By way of contrast, the neural network-based systems showed no degradation. MinDCF showed a similar characteristic.

Figure 3 shows the EERs for the DDAE-based system with respect to the weight parameter α of the cost function of the system, as expressed in Eq. (1). All values were measured on CC4 trials. Note that for $\alpha = 0$, the system is equivalent to the DAE-based system. It performs better than the DAE-based system for a wide range of α, with best performance achieved in the case of $\alpha = 0.5$.

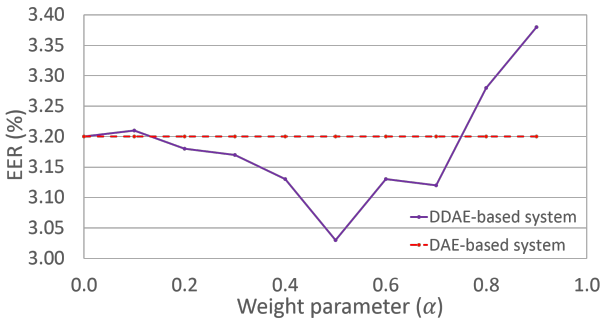

Figure 3: *Equal error rate (EER, %) of DDAE-based system with respect to weight parameter α. All values were measured on CC4. The DDAE-based system performs best in the case of $\alpha = 0.5$.*

Table 3: *Equal error rate (EER, %) and minimum detection cost function (minDCF) for different number of nodes in hidden layer of MLP section of DDAE-based system ($\alpha = 0.5$). All values were measured on CC4. The system performs best in the case of 2000 nodes.*

Number of nodes	EER%	minDCF
a) 1000	3.05	0.140
b) 2000	3.03	0.136
c) 3000	3.18	0.141

We also performed experiments on a number of nodes in the hidden layer of the MLP section in the DDAE-based system with respect to system performance in CC4 trials. The value of α was set to 0.5. Table 3 shows that the system with a hidden layer of 2000 nodes performed best.

Our proposed DDAE neural network performs better than multi-condition training in both clean and noisy conditions. DDAE also outperforms DAE because of the discriminative nature of denoised i-vectors, which contributes to improving speaker verification. These results indicate that neural networks such as DAE and DDAE are capable of reducing additive noise effects in i-vector space.

5. Conclusions and Future Work

In this paper, we have described novel discriminative denoising autoencoder-based i-vector transformation for handling additive noise in i-vector space in order to achieve robust speaker verification. The DDAE is trained to denoise and classify noisy i-vectors jointly, which adds discriminability in denoised i-vectors and consequently improves speaker verification under noisy conditions. We performed a series of experiments on a NIST SRE task to demonstrate the effectiveness of the transformations as compared to multi-condition training and DAE methods. Our proposed method outperformed the baseline system by 32% in EER as well as in minDCF for noisy test data. We have also shown that no degradation was observed in clean conditions, while multi-condition training increased error rates.

Our future work plans include comparison of denoising in i-vector space with that in acoustic feature space, as well as the application of our proposed idea to the handling of other critical issues in text-independent speaker verification systems, such as short utterances and channel distortion.

6. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[3] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[4] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[5] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 92–97.

[6] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos, "The ibm 2016 speaker recognition system," *arXiv preprint arXiv:1602.07291*, 2016.

[7] P. Matějka, O. Glembek, O. Novotnỳ, O. Plchot, F. Grézl, L. Burget, and J. H. Cernockỳ, "Analysis of dnn approaches to speaker identification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5100–5104.

[8] M. Kolbœk, Z.-H. Tan, and J. Jensen, "Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 305–311.

[9] N. B. Thomsen, D. A. L. Thomsen, Z.-H. Tan, B. Lindberg, and S. H. Jensen, "Speaker-dependent dictionary-based speech enhancement for text-dependent speaker verification," *Interspeech 2016*, 2016.

[10] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1741–1752, 2007.

[11] O. Plchot, L. Burget, H. Aronowitz, and P. Matjka, "Audio enhancing with DNN autoencoder for speaker recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5090–5094.

[12] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, "Towards noise-robust speaker recognition using probabilistic linear discriminant analysis," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 4253–4256.

[13] W. B. Kheder, D. Matrouf, J. F. Bonastre, M. Ajili, and P. M. Bousquet, "Additive noise compensation in the i-vector space for speaker recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4190–4194.

[14] Y. Z. Iik, H. Erdogan, and R. Sarikaya, "S-vector: A discriminative representation derived from i-vector for speaker verification," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug 2015, pp. 2097–2101.

[15] S. Novoselov, T. Pekhovsky, O. Kudashev, V. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," *ISCA Interspeech*, 2015.

[16] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Proc. Odyssey*, 2016.

[17] H. Yamamoto and T. Koshinaka, "Denoising autoencoder-based speaker feature restoration for utterances of short duration," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[18] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," *Interspeech 2016*, pp. 2369–2372, 2016.

[19] A. Narayanan and D. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2504–2508.

[20] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4375–4379.

[21] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[22] "The NIST year 2012 speaker recognition evaluation plan," https://www.nist.gov/multimodal-information-group/speaker-recognition-evaluation-2012, 2012.

[23] "The NIST year 2008 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/sre/2008/, 2008.

[24] "BOSARIS tool kit," http://sites.google.com/site/bosaristoolkit/.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[26] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[27] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot *et al.*, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of NIST 2011 workshop*. Citeseer, 2011.

[28] H. G. Hirsch, "FaNT - Filtering and Noise Adding Tool," http://dnt.kr.hsnr.de/download.html.

[29] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[30] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.