DNN Speaker Embeddings Using Autoencoder Pre-Training

Umair Khan and Javier Hernando
TALP Research Center, Department of Signal Theory and Communications,
Universitat Politecnica de Catalunya Barcelona, Spain
umair.khan@upc.edu, javier.hernando@upc.edu

***Abstract*—Over the last years, i-vectors have been the state-of-the-art approach in speaker recognition. Recent improvements in deep learning have increased the discriminative quality of i-vectors. However, deep learning architectures require a large amount of labeled background data which is difficult in practice. The aim of this paper is to propose an alternative scheme in order to reduce the need of labeled data. We propose the use of autoencoder pre-training in a speaker verification task. First, we train an autoencoder in an unsupervised way, using a large amount of unlabeled background data. Then, we train a Deep Neural Network (DNN) initialized with the parameters of the pre-trained autoencoder. The DNN training is carried out in a supervised way using relatively small labeled background data. In the testing phase, we extract speaker embeddings as the output of an intermediate layer of the DNN. The training and evaluation were performed on VoxCeleb-2 and VoxCeleb-1 databases, respectively. The experimental results have shown that by initializing DNN with the parameters of the pre-trained autoencoder, we have achieved a relative improvement of 21%, in terms of Equal Error Rate (EER), over the baseline i-vector/PLDA system.**

***Index Terms*—deep learning, autoencoders, i-vectors, speaker verification**

I. Introduction

The application of deep learning in speaker recognition is highly influenced by its success both in image and speech technologies [1, 2, 3, 4]. Deep learning applications in speaker recognition can be categorized in front-end and backend. As a front-end it is capable of learning deep features [5, 6, 7] and bottle neck features (BNF) that are used to compute Gaussian Mixture Models (GMM) posterior probabilities in a hybrid HMM-DNN model [8, 9]. Deep learning is also capable of learning speaker embeddings for speaker verification tasks such as in [10, 11, 12, 13]. As a backend, it is applied to improve the discriminative quality of i-vectors for speaker verification in [14, 15, 16].

In most of the deep learning approaches for classification tasks, it is required to train the network using a large amount of labeled data. The Probabilistic Linear Discriminant Analysis (PLDA) backend for i-vectors also requires labeled data. However, in practice, it is difficult to access large amount of labeled data, compared to unlabeled data. In this paper, we try to reduce the demand of labeled data in i-vector based speaker verification. Unsupervised deep learning approaches such as Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs) and autoencoders do not necessarily require labeled data. Several attempts have been made to improve the performance of speaker recognition using such unsupervised networks as in [17, 18, 19]. In [20, 19] a vector representation of speakers was proposed by means of RBMs in an unsupervised manner. These approaches rely on training a separate RBM model for every utterance in test data. Similarly, in [21], various algorithms are proposed to tackle the same problem using DBNs as a backend for i-vector based speaker verification. However, they rely on training a separate model for the target speaker only.

On the other hand, various alternatives of autoencoder training has been proposed for speaker recognition like in [22, 23]. In [24] the encoder part of an autoencoder has been used to extract speaker embeddings, that has been used for speaker segmentation. In [25] a denoising autoencoder has been trained to improve the signal quality. Similarly, in [26], autoencoder has been trained to learn a mapping between i-vectors with short and long utterances in speaker verification.

In this work, we propose the use of autoencoder pre-training for extracting DNN speaker embeddings from i-vectors in speaker verification task. In order to avoid the need of large amount of labeled data, we train the autoencoder using a large amount of unlabeled data. Then, we train a DNN classifier using a relatively small amount of labeled data. We propose to initialize the DNN training with the weight matrices and bias vectors of the pre-trained autoencoder. In this way, we train a hybrid autoencoder-DNN classifier. After the training, we extract speaker embeddings from i-vectors as the output form the second last layer of the network. The goal is to improve the performance using fewer background speaker labels. The experimental results have shown that the proposed approach has improved the baseline system in two aspects. Firstly, the proposed speaker embeddings, with cosine scoring, has gained a relative improvement of 21%, in terms of EER, over the baseline i-vector/PLDA system. Secondly, we have observed that the hybrid autoencoder-DNN training converges faster as compared to the one without autoencoder pre-training.

The rest of the paper is organized as follows. Section II explains the proposed method for DNN speaker embeddings extraction from i-vectors. Section III describes the experimental setup and database. The results obtained are discussed in Section IV. Finally, some conclusions are drawn in section V.

This work has been developed in the framework of the Camomile Project (PCIN-2013-067) and the Spanish Project DeepVoice (TEC2015-69266-P).

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

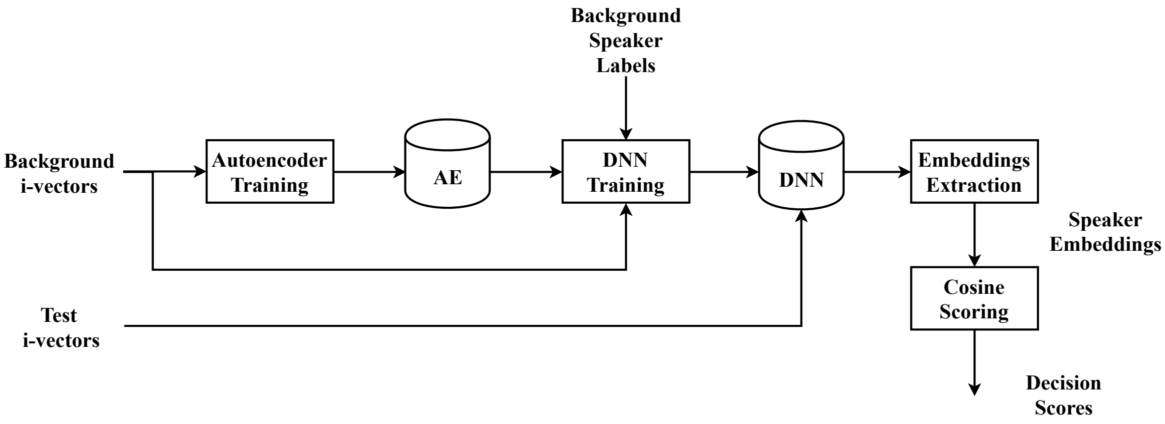


Fig. 1. Block diagram of the proposed speaker embeddings extraction from i-vectors using autoencoder pre-training.

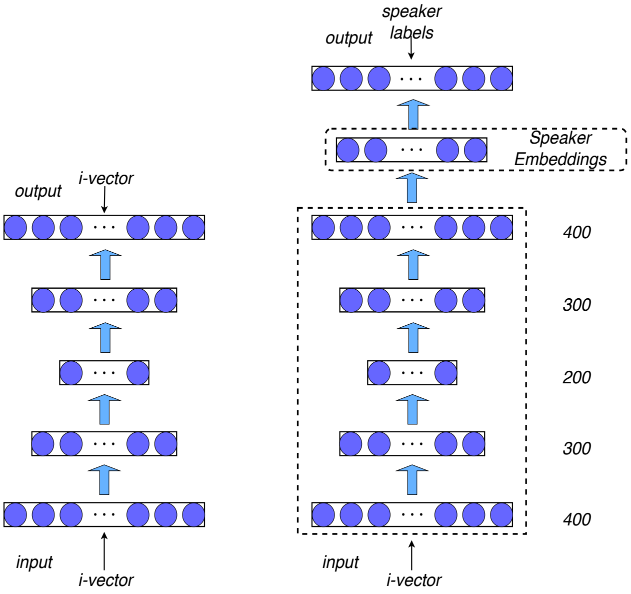


Fig. 2. (left) Autoencoder pre-training (right) DNN training.

II. Proposed Method

In this paper, we propose a new framework using autoencoder pre-training, to produce an alternative vector-based representation of speakers. Unlike DNN classifiers and i-vector/PLDA, we avoid using large amount of labeled data. Fig. 1 shows the block diagram of the proposed speaker embeddings extraction process. First, we train an autoencoder, in order to make use of the large amount of unlabeled background i-vectors. The training is carried out in a conventional way i.e., minimizing the Mean Square Error (MSE) loss between input and reconstructed i-vectors using the Stochastic Gradient Descent (SGD) optimizer. The autoencoder is supposed to learn speaker independent information from the background i-vectors as it has the ability to learn compact representation.

Once the autoencoder is trained, we train a DNN classifier in a supervised way, in order to learn information about speaker classes. We add a fully connected layer and a classification layer after the last layer of the autoencoder. We feed speaker labels at the output of the classification layer and train the network in a supervised manner. This network is referred to as hybrid autoencoder-DNN classifier and is trained using relatively smaller labeled i-vectors. We initialize the hybrid autoencoder-DNN with the weight matrices and bias vectors of the pre-trained autoencoder. This type of initialization has been applied for adapting the unsupervised model to learn speaker specific information in [18, 20]. The autoencoder pre-training helps in the supervised learning, and the network converges relatively faster than without pre-training.

There are two different scenarios in order to use the pre-trained autoencoder for the DNN initialization. One possibility is to add the fully connected and classification layers directly after the encoder part. The encoder part compresses the data into a shorter dimensional space which preserves enough information to reconstruct an approximation of the original data. However, this was not recommended in our experiments because in the hybrid autoencoder-DNN training the network learns additional information from the speaker labels fed at the output. The shorter dimensional space of the encoder part is not enough to learn efficient information from the higher dimensional classification layer.

Another scenario is to use the full autoencoder by adding the fully connected and classification layers at the end of the autoencoder. We prefer to expand the input data to its original dimensional space and then train the hybrid autoencoder-DNN classifier. In this way, firstly, we remove the unnecessary information from the data by encoding it into shorter dimensional space. Secondly, we expand the data back to its original dimensional space in order to ease the learning of additional information from the speaker labels.

Fig. 2 shows the architectures of both the autoencoder and the DNN. The encoder part and decoder part are symmetric as in a conventional autoencoder. The DNN has a similar structure except the fully connected and the output layers. Finally, we extract the output of the fully connected layer as the desired speaker embeddings that have shown to preserve speaker specific information. The test i-vectors are propagated

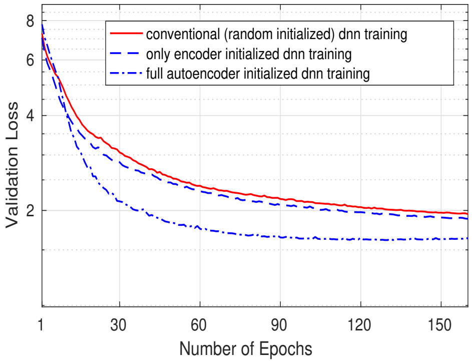


Fig. 3. Comparison of the training convergence, in terms of validation loss, between the conventional and the proposed training of the DNN classifier.

TABLE I
PERFORMANCE COMPARISON, IN TERMS OF EER (%), BETWEEN THE BASELINE AND THE PROPOSED SPEAKER EMBEDDINGS.

Approach	Scoring	EER(%)
i-vector	Cosine	17.61
i-vector	PLDA	9.54
only-encoder-dnn	Cosine	12.73
conventional-dnn	Cosine	8.58
full-autoencoder-dnn	Cosine	**7.51**

through the network in order to extract speaker embeddings from i-vectors. Using these embeddings, we perform the trials of the experiments with cosine scoring technique.

III. Experimental setup and database

The experiments were performed on VoxCeleb-1 and VoxCeleb-2 databases [27, 28] which contains 153,516 and 1,128,246 number of utterances, respectively. Both these databases are further partitioned into development and test sets. In this work, we have used the whole VoxCeleb-2 database (development and test) as unlabeled background data to train the autoencoder. The supervised training was carried out using the development partition of VoxCeleb-1 (the smaller database). VoxCeleb-1 is partitioned into 148,642 development and 4,874 test utterances, that belong to 1211 and 40 speakers, respectively. Thus, the classification layer in our hybrid autoencoder-DNN consists of 1211 number of neurons. From the test set of VoxCeleb-1, 37,720 experimental trials were scored. Half of them are target trials while the other half are non-target trials.

The development set of VoxCeleb-1 was used to train the Universal Background Model (UBM), the Total Variability (TV) matrix and the PLDA for the baseline i-vector/PLDA system. 20 dimensional MFCC features, appended by delta coefficients, were extracted for all the utterances. A 1024 components UBM is trained to extract 400 dimensional i-vectors. The PLDA for the i-vector/PLDA baseline was trained for 20 iterations and the number of eigenvoices was empirically set to 200. The UBM/TV matrix training and i-vector extraction process were carried out using Alize toolkit [29].

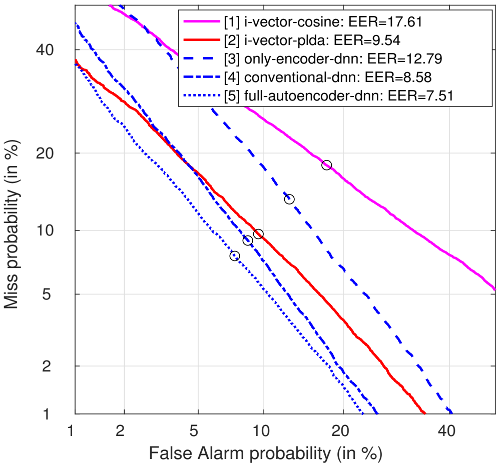


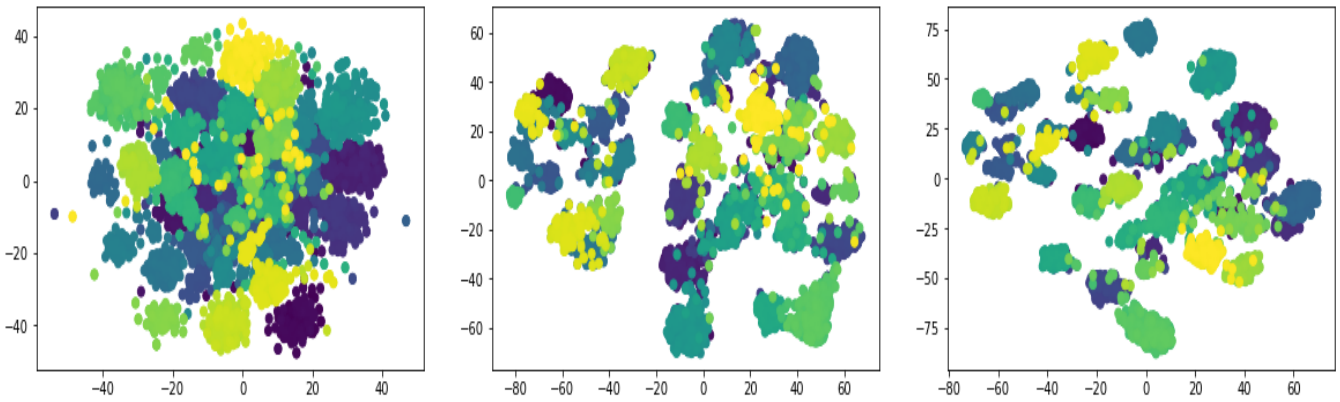
Fig. 4. DET plots of the baseline and the proposed speaker embeddings.

The autoencoder used in this paper consist of 3 hidden layers. The encoder and decoder parts are symmetrical and thus the hidden layer 1 and 3 have 300 neurons each, while hidden layer 2 consists of 200 neurons as shown in Fig. 2. The input and output layers consist of 400 neurons each. In the DNN, the dimension of speaker embeddings layer was fixed to 600 while the classification layer consists of 1211 neurons. The autoencoder pre-training was carried out for 400 epochs. All the layers of the autoencoder used ReLU activation function except the last layer which used linear activation. The learning rate was set to 0.03 with a decay of 0.0002 and the batch size was set to 100. The supervised DNN training was carried out for 200 epochs using Adagrad optimizer with an initial learning rate of 0.03 and a batch size of 100. Sigmoid activations were used for all the layers.

IV. Results

In the experiments it was observed that the autoencoder pre-training has learned speaker independent information from the large amount of unlabeled i-vectors. This information was utilized by the DNN classifier as it was initialized with the weights and biases of the autoencoder. This resulted in a significant improvement in the convergence of the DNN training. Fig. 3 shows the comparison of the DNN training i.e., conventional DNN and both the cases of autoencoder pre-trained DNN. In this paper, conventional DNN refers to randomly initialized DNN. The plots were obtained using the validation loss only. It can be seen that if the DNN was trained using only the encoder, as discussed in Section II, we have a slight improvement over randomly initialized DNN training. However the full autoencoder initialization is the best choice which helps in fast convergence of the training. For epochs greater than 10, the absolute value of the validation loss is reasonably lower as compared to the conventional training.

After extracting the desired speaker embeddings from i-vectors using the proposed approach, we score the experimental trials using cosine scoring technique. However the baseline



(a) Raw i-vectors

(b) Conventional DNN speaker embeddings

(c) Proposed speaker embeddings

Fig. 5. Comparison of the t-SNE Plots, between raw i-vectors, conventional and the proposed speaker embeddings. All the vectors were compressed to 2 dimensional space in order to generate the t-SNE plots.

i-vectors were scored using PLDA as well. Table I compares the performance of our proposed speaker embeddings with the conventional DNN speaker embeddings and i-vectors. Using only the encoder part to train the DNN, is not the preferred choice for our experiments. The full autoencoder initialization has an advantage of learning efficient information from the speaker labels as compared to the the only encoder case. From the table it is clear that our proposed speaker embeddings has outperformed both the other systems. The relative improvement between the proposed speaker embeddings and i-vector/PLDA is 21.28%, in terms of EER. If we compare the proposed speaker embeddings with the conventional DNN speaker embeddings, the relative improvement is 12.47%.

Fig. 4 shows the Detection Error Trade-off (DET) curves of our proposed approach and the other two approaches. We can see that the conventional DNN speaker embeddings perform worse than i-vector/PLDA, at low False Alarm (FA) regions. However, the DET plot for the proposed speaker embeddings shows better performance at all working regions.

Finally, in Fig. 5, we have shown the t-Distributed Stochastic Neighbor Embedding (t-SNE) plots for three different vector representation of speakers i.e., i-vectors, conventional DNN speaker embeddings and the proposed speaker embeddings. t-SNE is a dimensionality reduction technique to graphically visualize [30] higher dimensional vectors. In order to see the discriminative power of our proposed speaker embeddings, we have compared the t-SNE plots with the other two approaches. The plots were obtained using the test partition of the VoxCeleb-1 database. The dimensions of all the vectors was reduced to 2 for plotting the t-SNE. It is clear from the figure that our proposed speaker embeddings has the highest discrimination power as compared to the other two. The clusters generated are mostly pure and distinct. However, for the conventional DNN speaker embeddings, some of the clusters are overlapping with the others. For the baseline system, raw i-vectors were used to obtain the plots. The clusters formed for i-vectors are not very clear as compared to the former two DNN based speaker embeddings.

V. Conclusions

In this paper we proposed the use of autoencoder pre-training for DNN speaker embeddings in speaker verification task. The requirements of large amount of labeled data has put a constraint on deep learning approaches to this task. We put an effort to tackle this problem. In practical scenarios large amount of labeled data is not easily accessible. Thus we make use of unlabeled data to minimize the impact of lack of labeled data. In our proposed system, an autoencoder is pre-trained on a large amount of unlabeled background data which learns speaker independent information. Then a Deep Neural Network (DNN) classifier was trained using a relatively small labeled data, initialized with the parameters of the pre-trained autoencoder. For the experiments, speaker embeddings were extracted from i-vectors as the output of the embeddings layer of this autoencoder-DNN hybrid network. The evaluation was performed on the speaker verification trials of VoxCeleb-1 database. The results have shown that by using autoencoder pre-training for DNN, we gain a relative improvement of 21% in terms of EER, over the baseline i-vectors/PLDA system. Furthermore, we have observed that the DNN training converged faster, compared to the conventional (randomly initialized) dnn case.

References

[1] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 1695–1699.

[2] F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, Oct. 2015.

[3] K. Chen and A. Salman, “Learning speaker-specific characteristics with a deep neural architecture,” *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, Nov. 2011.

[4] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” in *Proc. Odyssey*, 2014, pp. 293–298.

[5] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[6] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, “Deep feature for text-dependent speaker verification,” *Speech Communication*, vol. 73, pp. 1–13, 2015.

[7] J. Jorrín, P. García, and L. Buera, “Dnn bottleneck features for speaker clustering,” *Proc. Interspeech 2017*, pp. 1024–1028, 2017.

[8] L. Deng, D. Yu, *et al.*, “Deep learning: Methods and applications,” *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[9] T. Yamada, L. Wang, and A. Kai, “Improvement of distant-talking speaker identification using bottleneck features of dnn.,” in *Interspeech*, 2013, pp. 3661–3664.

[10] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 4052–4056.

[11] Y. Z. Isik, 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)*, IEEE, 2015, pp. 2097–2101.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5329–5333.

[13] K. Okabe, T. Koshinaka, and K. Shinoda, “Attentive statistics pooling for deep speaker embedding,” *arXiv preprint arXiv:1803.10963*, 2018.

[14] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, “First attempt of boltzmann machines for speaker verification,” in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[15] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, “Preliminary investigation of boltzmann machine classifiers for speaker recognition,” in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[16] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, “Plda using gaussian restricted boltzmann machines with application to speaker verification,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[17] O. Ghahabi and J. Hernando, “Restricted boltzmann machines for vector representation of speech in speaker recognition,” *Computer Speech & Language*, vol. 47, pp. 16–29, Jan. 2018.

[18] O. Ghahabi and J. Hernando, “Deep belief networks for i-vector based speaker recognition,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 1700–1704.

[19] U. Khan, P. Safari, and J. Hernando, “Restricted Boltzmann Machine Vectors for Speaker Clustering,” in *Proc. IberSPEECH*, 2018, pp. 10–14.

[20] P. Safari, O. Ghahabi, and J. Hernando, “From features to speaker vectors by means of restricted boltzmann machine adaptation,” in *ODYSSEY 2016-The Speaker and Language Recognition Workshop*, 2016, pp. 366–371.

[21] O. Ghahabi and J. Hernando, “Deep learning backend for single and multisession i-vector speaker recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 807–817, Apr. 2017.

[22] 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,” *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[23] H.-S. Lee, Y.-D. Lu, C.-C. Hsu, Y. Tsao, H.-M. Wang, and S.-K. Jeng, “Discriminative autoencoders for speaker verification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 5375–5379.

[24] A. Jati and P. Georgiou, “Speaker2vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation,” *Proc. Interspeech 2017*, pp. 3567–3571, 2017.

[25] O. Novotny, O. Plchot, O. Glembek, J. Cernocky, L. Burget, *et al.*, “Analysis of dnn speech signal enhancement for robust speaker recognition,” *arXiv preprint arXiv:1811.07629*, 2018.

[26] O. Novotny, O. Plchot, P. Matejka, and O. Glembek, “On the use of dnn autoencoder for robust speaker recognition,” *arXiv preprint arXiv:1811.02938*, 2018.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” in *Interspeech*, 2017.

[28] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” in *Interspeech*, 2018.

[29] A. Larcher, J. F. Bonastre, B. G. B. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. D. Mason, and J. Y. Parfait, “Alize 3.0-open source toolkit for state-of-the-art speaker recognition.,” in *Interspeech*, 2013, pp. 2768–2772.

[30] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.