INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speaker Augmentation and Bandwidth Extension for Deep Speaker Embedding

Hitoshi Yamamoto, Kong Aik Lee, Koji Okabe, Takafumi Koshinaka

NEC Corporation, Japan

h-yamamoto@bc.jp.nec.com

Abstract

This paper investigates a novel data augmentation approach to train deep neural networks (DNNs) used for speaker embedding, i.e. to extract representation that allows easy comparison between speaker voices with a simple geometric operation. Data augmentation is used to create new examples from an existing training set, thereby increasing the quantity of training data improves the robustness of the model. We attempt to increase the number of speakers in the training set by generating new speakers via voice conversion. This speaker augmentation expands the coverage of speakers in the embedding space in contrast to conventional audio augmentation methods which focus on within-speaker variability. With an increased number of speakers in the training set, the DNN is trained to produce a better speaker-discriminative embedding. We also advocate using bandwidth extension to augment narrowband speech for a wideband application. Text-independent speaker recognition experiments in Speakers in the Wild (SITW) demonstrate a 17.9% reduction in minimum detection cost with speaker augmentation. The combined use of the two techniques provides further improvement.

Index Terms: speaker recognition, speaker embedding, data augmentation, bandwidth extension

1. Introduction

Speaker recognition is a task of recognizing the identity of a person, given a small amount of speech from the speaker [1, 2]. Recently, it has made remarkable progress with the use of deep speaker embedding [3, 4], a new speaker representation based on deep neural networks (DNNs). We propose a data augmentation method, referred to as speaker augmentation, for training such DNNs for extracting accurate deep speaker embedding.

Speaker embeddings are continuous-value vector representations that allow easy comparison between speaker voices with a simple geometric operation. Among others, i-vector [5] and x-vector [4] emerged as main-stream methods for speaker embedding. An i-vector captures the deviation of a speech utterance in comparison with a background model. It utilizes a Gaussian mixture model (GMM) to measure such differences in the acoustic space and compresses them into a single vector. The x-vector, however, represents speakers in the speaker space by using a DNN trained with a speaker discriminative cost.

Data augmentation is a procedure to generate artificial training examples from existing training data. It is widely used in machine learning to prepare a large training set with a vast diversity of examples [6]. With the widespread use of deep learning, data augmentation has become an indispensable step in system development. Data augmentation methods have been extensively investigated for audio and speech recognition tasks, e.g., ASR [7, 8, 9], speaker recognition [10, 11], acoustic events [12] and scene [13] classification, and music processing [14]. Notice that these prior studies aimed to create training examples of existing classes to cover within-class variability.

In contrast to conventional audio augmentation, our speaker augmentation aims to increase the number of speakers from an existing training set. Our motivation is to use more speakers for training the DNN to obtain an accurate speaker-discriminative feature representation. In [15], it was shown that the number of speakers is an important factor for good performance. Such observation is also consistent with that reported in face recognition, where millions of identities are typical for training [16].

We use voice conversion for speaker augmentation. Our aim is different from those in ASR, where techniques like vocal tract length perturbation (VTLP) [17], stochastic feature mapping (SFM) [18], and speed perturbation [19] are used to increase the quantity of data while keeping the labels unchanged. Also different from that in [20], we show that speed perturbation (and other voice conversion techniques) generate new voices and therefore new speaker labels have to be assigned to them. This is critical for speaker recognition task, where classes form for speakers instead of senones. The main contribution of this paper is a data augmentation method for helping to collect new speakers at the lowest cost and thereby improve the robustness of speaker embedding.

We also investigate mixed-bandwidth training as another strategy to increase the number of speakers in a training set. It involves using multiple training sets to train a single DNN. In ASR, this is achieved by designing acoustic features to be shared between narrowband and wideband speech [21]. A similar data mixing method has also been investigated in speaker recognition [22]. We implement a new pipeline that uses a DNN-based bandwidth extension (BWE) [23] as pre-processing of the DNN for speaker embedding extraction. The BWE generates missing frequency bands of narrowband speech from its low-band information. The second contribution of this paper is a comparison of speaker augmentation and bandwidth extension with respect to the number of speakers in training data. Moreover, the combination of the two methods is evaluated. It is worth mentioning that all these methods for increasing speakers have attracted much attention during the SRE18 workshop.

This paper is organized as follows: Section 2 introduces deep speaker embedding; Section 3 presents our speaker augmentation that generates training example of new speakers; Section 4 explains mixed-bandwidth training; Section 5 describes experimental evaluation results for speaker verification in SITW and NIST SRE tasks, and Section 6 summarizes our work.

2. Deep Speaker Embedding

This section briefly presents deep speaker embedding which is widely used in state-of-the-art speaker recognition systems. In particular, we use x-vector [4] shown in Figure 1. A segment-level x-vector $\boldsymbol{y}$ is extracted from a sequence of acoustic fea-

http://dx.doi.org/10.21437/Interspeech.2019-1508

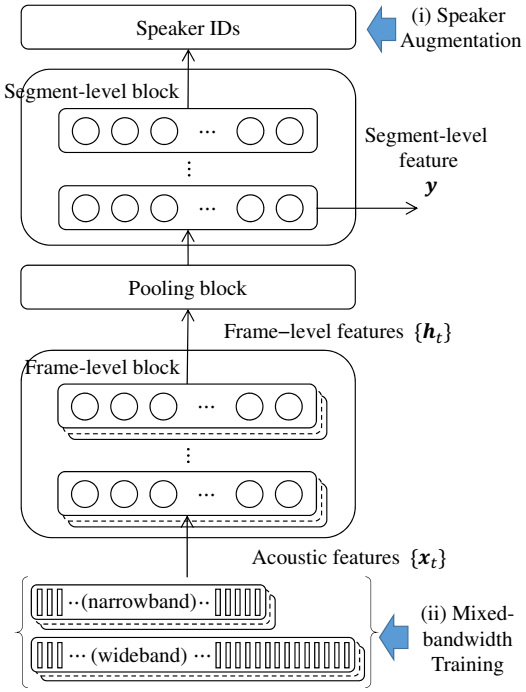

Figure 1: *DNN for extracting deep speaker embedding. A segment-level speaker representation $\boldsymbol{y}$ is computed from a sequence of acoustic features $\{\boldsymbol{x}_t\}$.*

tures $\{\boldsymbol{x}_t\}$ using a deep neural network (DNN). The DNN consists of three components, frame-level, pooling, and segment-level block.

The frame-level block input a sequence of acoustic features $\{\boldsymbol{x}_t\}$, e.g., MFCCs. After considering a relatively short-term context, this block outputs frame-level features $\{\boldsymbol{h}_t\}$. A Time-Delay Neural Network (TDNN) is used for this block [4]. Another type of DNN, convolutional or recurrent one, is applicable as known in other speaker embedding studies [24, 25].

The pooling block converts arbitrary number of frame-level features $\{\boldsymbol{h}_t\}$ into a single fixed-dimensional vector. In case of x-vector, its statistic pooling layer aggregates all frame-level features and computes their mean and standard deviation. This block can include an additional attention mechanism to give different weight for each frames [11].

The segment-level block maps the segment-level vector to speaker identities (IDs). One of the layers is designed as bottleneck layer, which forces the information brought from the preceding layer into a low-dimensional representation. Then we can use such bottleneck features as segment-level features $\boldsymbol{y}$.

The DNN is trained to classify the N speakers in a given training set. Each of the nodes in the output layer corresponds to one of the N speaker ID. This paper focuses on increasing the number of speakers to $N' > N$ with (i) speaker augmentation and (ii) mixed-bandwidth training.

3. Speaker Augmentation

We compare speaker augmentation with conventional audio augmentation. Figure 2 illustrates how the two methods augment a training set. As shown on the left, audio augmentation creates examples from a source example. It assigns the source speaker ID to the augmented examples since it takes into ac-

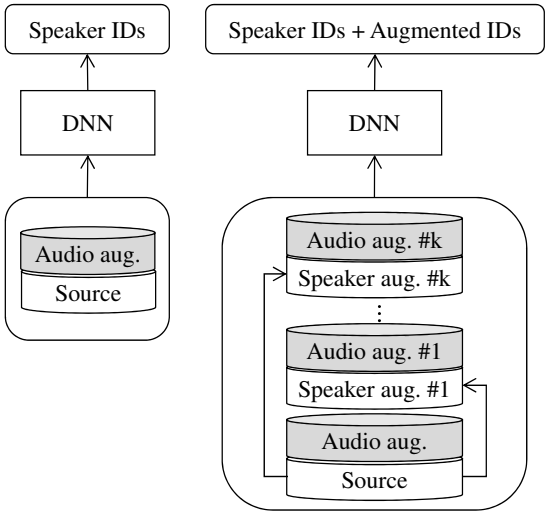

Figure 2: *Data augmentation for training set of DNNs for deep speaker embedding. Left: Conventional audio augmentation. Right: Proposed speaker augmentation that generates new speaker's examples from source speakers.*

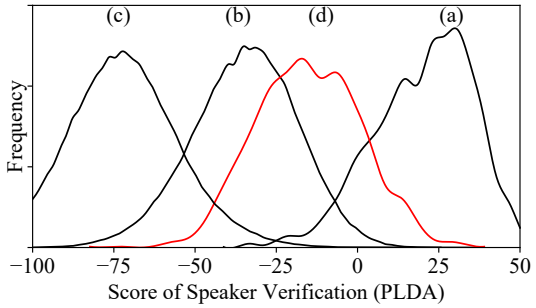

Figure 3: *Score distributions for (a) same speaker, (b) different speakers, (c) different genders and (d) same speaker with speaker augmentation on test-side. Scores of target trials decreased after speaker augmentation.*

count both the augmented and source examples spoken by the same speaker. We followed Kaldi's x-vector recipe [10] which artificially corrupts speech with additive noise (babble, music, noise) and reverberant noise (room impulse responses).

Speaker augmentation creates additional target speakers of a DNN for speaker embedding. As shown on the right of Figure 2, it generates examples from a source example and assigns a new speaker ID to the augmented examples since it takes into account both the augmented and source examples belonging to different speakers. We can then increase N speaker IDs in the training set. For instance, by making k copies of each source example, $N' = (k+1)N$ speaker IDs are available for DNN training. We use speed perturbation [19] based on the SoX [26] *speed* function that modifies the pitch and tempo of speech by resampling. Note that, after speaker augmentation, it is possible to apply conventional audio augmentation to the augmented examples.

We conducted a preliminary experiment to look into how speaker embedding changes by speaker augmentation. Figure 3 illustrates the score distributions (normalized histograms) from our baseline speaker verification system (see 5.1 for details) for

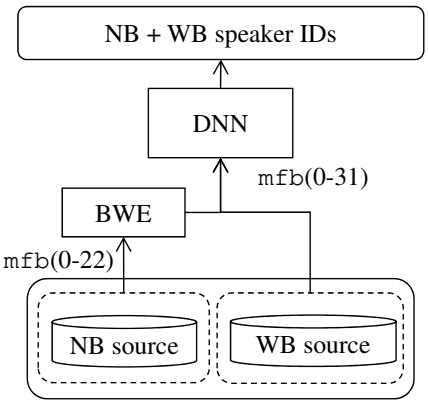

Figure 4: *Mixed-bandwidth training for training set of DNN for deep speaker embedding. Common acoustic feature (MFB) are extracted from both wideband (WB) and narrowband (NB) speech with the use of DNN of bandwidth expansion (BWE).*

four trial lists. Three lists were taken from the Speakers in the Wild evaluation set (SITW-eval), (a) *target* (same speaker), (b) *non-target* (different speakers) and (c) *cross-gender*. Another trial list (d) was created based on the *target* by applying speed perturbation only to the test-side speech (speed factor: 0.9). We can see that the scores of (d) are lower than *target* and the peak of distribution (d) is near to that of *non-target* rather than that of *target*. This result indicates that many speakers after the modification have different x-vectors from the source speakers. Note that the score distribution of the *non-target* trials with speed perturbation only to the test-side speech is similar to (b). Therefore, it should be reasonable to assign new speaker labels to the augmented examples.

4. Mixed-bandwidth training

This section explains mixed-bandwidth training for x-vector extraction. We investigate the combined use of narrowband and wideband corpora to increase not only the quantity of training data but also the number of speakers. A straightforward option is narrowband training where wideband speech (e.g. 16kHz) is down-sampled (e.g. to 8kHz) and merged into the narrowband set. Another approach we investigate is mixed-bandwidth training in which both narrowband and wideband speech are mapped into common acoustic features.

We consider making use of log mel-filterbank coefficients (MFBs) to share acoustic features between narrowband and wideband speech. It is possible to design such filterbanks by tuning cut-off frequencies. An example of a configuration used in our evaluation is as follows; 23 filterbanks for narrowband speech ($20 - 3700$ Hz) and 32 filterbanks for wideband speech ($20 - 7974$ Hz). In this case, narrowband filterbanks are approximately aligned with wideband ones (0th to 22nd).

In mixed-bandwidth training, MFBs corresponding to a higher frequency have zero values for narrowband speech. Even in such a case, DNN improves ASR accuracy since it has the flexibility to accept missing features as shown in the previous study [21]. However, there is still a gap from the ideal case using the same amount of wideband training data as the narrowband data. Therefore, we develop a pipeline of x-vector extraction including a pre-processing DNN, which takes the role of BWE to compensate for such missing MFBs. As illustrated in Figure 4, the DNN for BWE inputs 23 MFBs ($0 - 22$), estimates 9 MFBs ($23 - 31$) for higher frequency and concatenates them to form 32 MFBs ($0-31$). We assume this is good for narrowband speech since the original 23 MFBs remain unchanged. The succeeding DNN for x-vector can be trained with common 32 MFBs for both bands.

5. Evaluation

We evaluated the performance of the speaker augmentation and mixed-bandwidth training in text-independent speaker verification tasks of Speakers in the Wild (SITW) [27] and NIST 2016 [28] and 2018 [29] Speaker Recognition Evaluation (SRE). Performance measures for the evaluation were the minimum detection costs (mDCFs) and the equal error rates (EERs) on the evaluation set.

5.1. Speaker augmentation on wideband speech

The overall procedure of speaker verification was based on the `sitw/v2` recipe in Kaldi [30]. Acoustic features were 30-dimensional MFCCs extracted at every 10ms, which are mean-normalized within a 3-second-width sliding window and then segmented with energy-based voice activity detection (VAD). The x-vectors were computed from the acoustic features with a DNN which consists of a TDNN with 5 layers, a statistic pooling layer, and 2 fully connected layers. Further, these x-vectors were centered, compressed by LDA, and length-normalized. Verification scores were computed using a PLDA with a speaker space of 150 dimensions.

A baseline system WB was trained with VoxCeleb 1 [24] and 2 [31], a collection of YouTube videos including interviews of celebrities. We concatenated all clips from the same video into one file to make the data more appropriate for training as suggested in [32] (VoxCelebCat). The number of speakers and recordings were 7,185 and 294,600, respectively. Our system WB+Aug was trained with additional examples created by the speed perturbation, including k ($k = 1, 2$) copies for each of the source example (speed factor: 0.9 and 1.1). An evaluation set was SITW, a collection of speech across a wide array of challenging conditions. We used the SITW-eval core-core trial list, including 721,788 pairs of single-speaker audio recordings.

Table 1 shows the results of wideband systems on the SITW-eval core-core trial list. The mDCFs with our speaker augmentation (WB+Aug) were 0.282 and 0.271 and outperformed the 0.330 of the baseline WB. The errors were reduced by increasing the number of speakers to 14,370 and 21,555. For WB+Aug(0.9, 1.1), the reduction rate of mDCF and EER were 17.9 and 16.8%, respectively. These results indicate that speaker augmentation provides informative speakers to train speaker-discriminative DNNs. Note that our baseline was better than `sitw/v2` ($\mathrm{mDCF}_{0.01}$: 0.342, EER: 3.50%). This might be due to the VoxCelebCat training set.

We analyzed the effectiveness of speaker augmentation for different settings of iteration times during DNN training in Table 2 (#iter) since the Kaldi's script determines #iter in proportion to the size (number of frames) of the training set. To conduct this analysis, we increased the number of epochs for WB (3 to 9). For WB+Aug, we reduced the size of the training set to the same for WB. There was a consistent improvement with speaker augmentation for all the iteration times.

5.2. Speaker augmentation on narrowband speech

These systems were configured for narrowband speech, based on the `sre16/v2` recipe in Kaldi. Differences from the

Table 1: *Minimum detection costs (mDCF) and equal error rates (EER, %) for wideband systems on SITW-eval core-core trials. Suffix of mDCF means P_{Target}, a priori probability of the specified target speaker. Speaker augmented systems (WB+Aug) outperformed the baseline (WB).*

System	#ID	$\mathrm{mDCF}_{0.01}$	EER(%)
WB	7185	0.330	3.28
WB+Aug(0.9)	14370	0.282	2.98
WB+Aug(0.9, 1.1)	21555	0.271	2.73

Table 2: *Comparison of wideband systems with respect to iteration times (#iter) in training on SITW-eval core-core trials. Speaker augmentation showed a consistent improvement.*

System	#iter	$\mathrm{mDCF}_{0.01}$	EER(%)
WB	45	0.330	3.28
WB+Aug(0.9, 1.1)	45	0.300	2.98
WB	135	0.303	3.09
WB+Aug(0.9, 1.1)	136	0.271	2.73

wideband system in 5.1 were using 23-dimensional MFCCs as acoustic features and applying domain adaptation to PLDA.

A baseline system NB was trained with English telephone conversations including SRE04-10, Mixer6, Switchboard2, and Switchboard Cellular. The number of speakers and recordings in total was 5,145 and 211,070 in total, respectively. Our system NB+Aug was trained with additional examples by speed perturbation as WB+Aug. In addition, we made MB(8k) which incorporated VoxCeleb down-sampled to 8kHz to conduct data mixing in narrowband. We used SRE16 and SRE18 for evaluation. The SRE16 includes 1,986,728 trials taken from *call-my-net* (CMN) telephone conversation spoken in Cantonese or Tagalog. The SRE18 consists of 2,094,823 trials taken from *call-my-net2* (CMN2) spoken in Tunisian Arabic.

Table 3 shows the accuracy of narrowband systems. The performance metric here is a minimum cost ($C_{primary}$), the average of two DCFs, defined for SRE16 [28]. Similar to the wideband case, speaker verification improved in proportion to the number of speakers in the training set. Another interesting result is that speaker augmentation (NB+Aug) was comparable to data mixing in narrowband (MB(8k)). This suggests that virtual speakers generated using speaker augmentation are just as effective in assisting the training as real speakers incorporated from other corpora.

5.3. Mixed-bandwidth training

These systems were designed to compare speaker augmentation with mixed-bandwidth training. A baseline system was the same as the wideband system WB in 5.1 except it used 32-dimensional MFBs and original VoxCeleb for training. MB(16k) was based on conventional mixed-bandwidth training with zero-padding [21]. It combined VoxCeleb and the telephone corpora in 5.2. MB+BWE further had DNN-based BWE trained with VoxCeleb as the pre-processing of x-vector extraction. The BWE consists of 5 fully connected layers with an input size of 23×5 (i.e., a context of ± 2), and the output size is 9. WB+Aug was WB with speaker augmentation, using 3-times larger number of speaker IDs and examples for training. MB+BWE+Aug was based on the combination of the two

Table 3: *Minimum $C_{primary}$ metrics for narrowband systems on SRE16 and SRE18. Speaker augmentation (NB+Aug) was comparable to data mixing in narrowband MB(8k).*

System	#ID	#iter	SRE16	SRE18
NB	5145	81	0.590	0.551
NB+Aug(0.9)	10290	81	0.573	0.524
NB+Aug(0.9, 1.1)	15435	81	0.547	0.518
MB(8k)	12330	81	0.568	0.519

Table 4: *Results for SRE18 systems with speaker augmentation and mixed-bandwidth training with bandwidth extension on SITW-eval core-core trials. In SRE18,* $\mathrm{mDCF}_{0.05}$ *was used for wideband trials.*

System	#ID	$\mathrm{mDCF}_{0.05}$	EER(%)
WB	7185	0.213	3.14
MB(16k)	12330	0.180	2.73
MB+BWE	12330	0.178	2.62
WB+Aug	21555	0.170	2.54
MB+BWE+Aug	27200	0.151	2.18

methods. Note that it was developed for SRE18 [29] and had differences from the others. It extended the x-vector to use a multi-head attentive pooling mechanism [11] and updated the DNN progressively by adding new augmented data [33, 34]. Although a fair comparison is difficult because of the difference, it shows a case of making the best effort to enlarge the number of speaker IDs within these experiments for SRE18.

Table 4 shows that both speaker augmentation (WB+Aug) and mixed-bandwidth training with bandwidth extension (MB+BWE) outperformed the baseline (WB). The BWE resulted in a slight improvement in EER compared with standard mixed-bandwidth training (MB(16k)). MB+BWE+Aug was the most accurate with the combination of the speaker augmentation and BWE. Here, the reduction rate of mDCF and EER from MB+BWE were 15.1 and 16.8%, respectively. The additional attentive pooling covered half of the gain. These results indicate that the two sets of augmented speakers from speaker augmentation and bandwidth extension have complementary information for speaker-discriminative DNNs.

6. Conclusions

We have presented speaker augmentation and mixed-bandwidth training with bandwidth extension for deep speaker embedding. Our speaker augmentation applies speed perturbation to source examples to generate new speakers' examples. DNN-based bandwidth extension augments narrowband speech for wideband, by generating missing features in higher frequency. The deep speaker embedding, x-vector, becomes accurate by increasing the number of speakers for training speaker-discriminative DNN. The effectiveness of the methods was demonstrated in a series of text-independent speaker verification experiments on SITW and NIST SRE tasks. Speaker augmentation achieved 17.9 and 16.8% reduction in minimum DCF and EER on SITW over those for a case without augmentation. The combination of the two methods showed further improvement. Pursuing even better techniques for speaker augmentation is an issue for our future work.

7. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[2] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74 – 99, 2015.

[3] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[5] N. Dehak, P. 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, 2010.

[6] B. Sapp, A. Saxena, and A. Y. Ng, "A fast data collection and augmentation procedure for object recognition," in *Proc. AAAI*, 2008.

[7] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. ASRU*, 2013.

[8] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv e-prints*, p. arXiv:1412.5567, Dec 2014.

[9] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, 2015.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

[11] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[12] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Proc. Interspeech*, 2016.

[13] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Letter*, vol. 24, no. 3, pp. 279–283, 2017.

[14] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proc. ISMIR*, 2015, pp. 248–254.

[15] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *SLT*, 2016, pp. 165–170.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015.

[17] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[18] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *Proc. ICASSP*, 2014.

[19] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015.

[20] M. Mclaren, D. Castán, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Proc. Odyssey*, 2018, pp. 327–334. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2018-46

[21] J. Li, D. Yu, J.-T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CDDNN-HMM," in *Proc. SLT*, 2012, pp. 131–136.

[22] P. Nidadavolu, C.-I. Lai, J. Villalba, and N. Dehak, "Investigation on bandwidth extension for speaker recognition," in *Proc. Interspeech*, 2018.

[23] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. Interspeech*, 2015.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A largescale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[25] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.

[26] "SoX – Sound eXchange." [Online]. Available: http://sox.sourceforge.net/

[27] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Interspeech*, 2016.

[28] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," in *Proc. Interspeech*, 2017, pp. 1353–1357.

[29] S. O. Sadjadi, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Proc. Interspeech (submitted)*, 2019.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[31] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018.

[32] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. P. Garcia-Perera, P. A. Torres-Carrasquillo, and N. Dehak, "The JHU-MIT system description for NIST SRE18," *NIST SRE 2018 Workshop*, 2018.

[33] K. A. Lee, H. Yamamoto, K. Okabe, Q. Wang, L. Guo, T. Koshinaka, J. Zhang, and K. Shinoda, "The NEC-TT speaker verification system for SRE18," *NIST SRE 2018 Workshop*, 2018.

[34] ——, "The NEC-TT 2018 speaker verification system," in *Proc. Interspeech*, 2019.