CAM: CONTEXT-AWARE MASKING FOR ROBUST SPEAKER VERIFICATION

Ya-Qi Yu[1*], *Siqi Zheng*[2], *Hongbin Suo*[2], *Yun Lei*[2], *Wu-Jun Li*[1]

[1]National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
[2]Speech Lab, Alibaba Group

yuyq@lamda.nju.edu.cn, {zsq174630, gaia.shb, yun.lei}@alibaba-inc.com, liwujun@nju.edu.cn

ABSTRACT

Performance degradation caused by noise has been a long-standing challenge for speaker verification. Previous methods usually involve applying a denoising transformation to speaker embeddings or enhancing input features. Nevertheless, these methods are lossy and inefficient for speaker embedding. In this paper, we propose context-aware masking (CAM), a novel method to extract robust speaker embedding. CAM enables the speaker embedding network to "focus" on the speaker of interest and "blur" unrelated noise. The threshold of masking is dynamically controlled by an auxiliary context embedding that captures speaker and noise characteristics. Moreover, models adopting CAM can be trained in an end-to-end manner without using synthesized noisy-clean speech pairs. Our results show that CAM improves speaker verification performance in the wild by a large margin, compared to the baselines.

Index Terms— Speaker verification, speech enhancement, context embedding, context-aware masking

1. INTRODUCTION

The task of speaker verification is to verify speaker identities by comparing test against enrollment utterances. Generally, speaker verification includes two steps. The first step is to extract speaker embedding, which maps utterances of various lengths into fixed-dimensional vectors. The second step is to score the speaker consistency of embeddings. In the past few years, deep learning-based speaker embedding has made significant progress. Speaker verification systems utilizing deep learning-based speaker embedding have shown state-of-the-art performance [1, 2, 3, 4, 5, 6].

Despite the progress mentioned above, performance degradation caused by noise remains a challenge for speaker embedding. In terms of model design, previous efforts to tackle this challenge usually follow one of the two methodologies below. The first methodology is to apply a denoising transformation to speaker embeddings. In [7, 8, 9], researchers use either statistical back end or neural network back end to transform noisy speaker embeddings into enhanced ones. A problem with this methodology is information loss. Generally, speaker embeddings are extracted by aggregating hundreds of frame-level features into one utterance-level embedding through a statistical pooling layer such as mean pooling. Information loss is inevitable at the aggregation step. Hence, it is lossy to post-process noisy speaker embeddings after the aggregation step.

The second methodology is to filter noise through a speech enhancement model and extract robust speaker embedding based on the enhanced features. Classical speech enhancement methods usually adopt supervised training [10]. These methods train models based on noisy-clean speech pairs, which are synthesized data and not strictly consistent with real distributions. Researchers also propose methods to adopt generative adversarial networks (GANs) for unsupervised speech enhancement [11, 12, 13]. Nevertheless, these methods are not task-specific, and the training data of speech enhancement and speaker embedding might belong to different domains. In [14], researchers evaluate and optimize the speech enhancement model based on perceptual loss, which is calculated by a pre-trained speaker embedding network. In [15, 16, 17], researchers connect and train the speech enhancement and speaker embedding networks in an end-to-end manner. Both perceptual loss and end-to-end training optimize the speech enhancement network with the target of improving speaker verification performance instead of decreasing the regression error between enhanced and clean features. With an unified target during training and inference, these methods are supposed to obtain better performance on the specific task, i.e., speaker verification.

Nevertheless, the practice of connecting speech enhancement and speaker embedding models might be redundant for three reasons. First, it is lossy and inefficient to filter noise in the raw feature space, i.e., waveform or spectrogram. Second, the speech enhancement model is sensitive to training data and massively increases the computational cost. Third, it is possible to empower the speaker embedding model to handle noise, just with a few modifications.

Hence, we propose to adopt a different methodology for extracting robust speaker embedding. First, the denoising step should be earlier than the aggregation step to avoid information loss. Second, we do not use a separate speech enhancement model to pre-process input features. This methodology allows us to modify the speaker embedding backbone.

The contributions of this paper are listed as follows:

- We propose context-aware masking (CAM), a novel method to extract robust speaker embedding. CAM empowers the speaker embedding network to focus on the speaker of interest and blur unrelated noise. Compared to previous speech enhancement-based methods, CAM requires only a few percent of the computational cost.
- We propose the idea of extracting an auxiliary context embedding that captures the speaker and noise characteristics. We prove that auxiliary context embedding is an essential component in CAM.

The rest of this paper is organized as follows. Section 2 introduces two related methods. Section 3 introduces CAM and demonstrates it with two speaker embedding backbones. Section 4 shows the experiment and visualization results. Section 5 is the conclusion.

[*] The work was done during an internship at Alibaba.

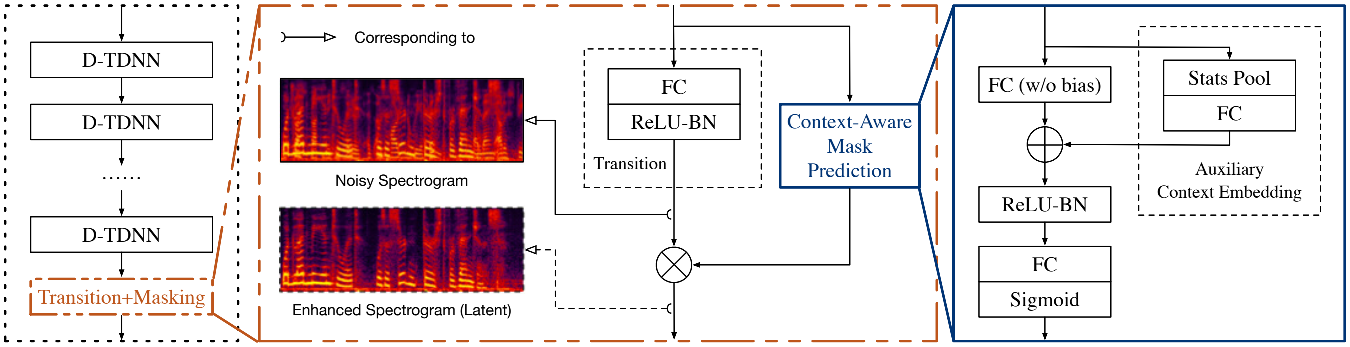

Fig. 1. Applying CAM to the speaker embedding backbone (D-TDNN). (a) Left: A D-TDNN block which consists of several D-TDNN layers and a transition layer. (b) Center: The transition layer and feature map masking. (c) Right: The context-aware mask prediction module.

2. RELATED WORKS

In this section, we briefly introduce some related methods, including context-aware speech enhancement and attentive statistics pooling.

2.1. Context-Aware Speech Enhancement

Speaker-aware speech enhancement considers the speaker characteristics while enhancing the input features. Recently, researchers in [18, 19, 20] utilize reference speaker embedding to discriminate the speaker of interest during enhancement. The reference speaker embedding can be extracted from the input utterance or another clean utterance. Similarly, there is a related technique called noise-aware training [21, 22] for speech enhancement. On the opposite of speaker-aware speech enhancement, noise-aware training makes use of the noise characteristics. For example, the noise characteristics can be embedded in a dynamic noise embedding extracted from non-speech frames [22]. In summary, both techniques are aware of the global context, either the target context (i.e., speaker) or non-target context (i.e., noise).

2.2. Attentive Statistics Pooling

Attentive statistics pooling (ASP) [23] is an attention-based pooling strategy. The motivation of ASP is to assign different weights to different frames, e.g., to give noisy frames small weights. ASP adopts a simple module to calculate the attention weights:

$$s_t = \mathbf{v}^\top \delta(\mathbf{U}^\top \mathbf{h}_t + \mathbf{p}) + q, \quad (1)$$

$$\alpha_t = \frac{\exp(s_t)}{\sum_{\tau=1}^{T} \exp(s_\tau)}, \quad (2)$$

where $\delta(\cdot)$ denotes the activation function, such as tanh. $\mathbf{h}_t$, s_t and α_t denote the hidden feature vector, score and attention weight at the t-th frame, respectively.

3. PROPOSED METHODS

In this section, we propose CAM and demonstrate it with two time-delay neural network (TDNN) [1] based speaker embedding backbones. Figure 1 is an example of CAM. The left box shows a part of the speaker embedding backbone, in which we equip the transition layer with a masking mechanism. The center box shows the detailed masking method. The right box shows the detailed method for predicting context-aware mask.

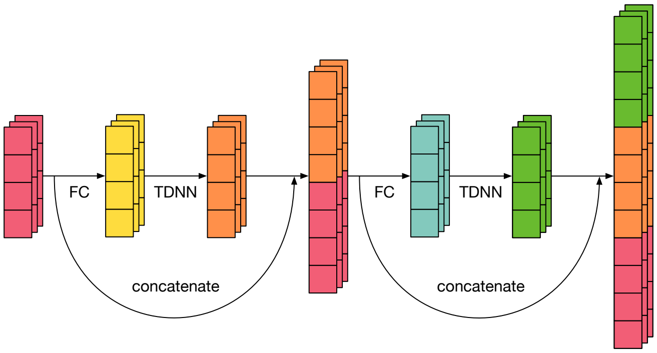

Fig. 2. Example of two consecutive D-TDNN layers.

3.1. Speaker Embedding

TDNN, also known as dilated one-dimensional convolution (Dilated 1D Conv), has been widely used in speaker verification tasks [1, 4, 5, 6]. In [1], researchers propose a TDNN-based model for extracting speaker embeddings. We denote this speaker embedding model as the vanilla TDNN and adopt it as the first speaker embedding backbone. Specifically, the vanilla TDNN consists of three consecutive TDNN layers, followed by two consecutive position-wise fully connected (FC) layers, a statistics pooling layer, and an FC-based embedding layer.

Recently, we propose another TDNN-based speaker embedding model named densely connected TDNN (D-TDNN) [5]. D-TDNN achieves state-of-the-art performance on the popular speaker verification dataset VoxCeleb [24] combined with cosine scoring. At the same time, D-TDNN contains fewer parameters compared to previous models. Hence, we adopt D-TDNN as a representative of the state-of-the-art speaker embedding models.

The basic unit of D-TDNN, named the D-TDNN layer, mainly consists of a position-wise FC-based bottleneck layer and a TDNN layer. We concatenate the input and inner output of the D-TDNN layer to form its final output. Figure 2 shows the major transformations of feature maps in two consecutive D-TDNN layers. Our experiment follows the network settings in [5] except that we use a structure of TDNN-ReLU-BatchNorm (BN) to replace BN-ReLU-TDNN since we find the former is more memory friendly. To be specific, D-TDNN has *two* blocks. The first block contains 6 D-TDNN layers, and the second one contains 12 D-TDNN layers.

3.2. Context-Aware Masking

We denote the input acoustic features as $\mathbf{x}$. The output feature map of a selected hidden layer is:

$$g(\mathcal{F}(\mathbf{x})), \tag{3}$$

where $g(\cdot)$ denotes the transformation of the hidden layer and $\mathcal{F}(\cdot)$ denotes the transformation of the layers before the hidden layer.

In previous speech enhancement-based methods, the first step is to predict enhanced acoustic features, denoted as $\tilde{\mathbf{x}}$. The corresponding output feature map of the hidden layer becomes:

$$g(\mathcal{F}(\tilde{\mathbf{x}})). \tag{4}$$

As shown in Figure 1 (b), feature map masking is to apply a ratio mask on the feature map. It is possible to obtain a similar effect with speech enhancement if we can find a mask $\tilde{\mathbf{M}}$ that satisfies:

$$g(\mathcal{F}(\mathbf{x})) \odot \tilde{\mathbf{M}} \propto g(\mathcal{F}(\tilde{\mathbf{x}})), \tag{5}$$

where $\odot$ denotes element-wise multiplication, and the elements of $\tilde{\mathbf{M}}$ can be normalized to the range of $(0, 1)$.

In the following part, we demonstrate how to estimate a proper ratio mask. To achieve this goal, we employ the characteristics of the speaker of interest and noise, which can be derived from an auxiliary context embedding, denoted as $\mathbf{e}$. As shown in Figure 1 (c), we predict the context-aware mask frame by frame:

$$\mathbf{F} = \mathcal{F}(\mathbf{x}), \tag{6}$$

$$\mathbf{M}_{*t} = \sigma(\mathbf{W}_2^\top \omega(\mathbf{W}_1^\top \mathbf{F}_{*t} + \mathbf{e}) + \mathbf{b}_2), \tag{7}$$

$$\tilde{\mathbf{F}} = g(\mathcal{F}(\mathbf{x})) \odot \mathbf{M}, \tag{8}$$

where $\sigma(\cdot)$ denotes the Sigmoid function, $\omega(\cdot)$ denotes the combination of ReLU function and BN. $\mathbf{M}$ is the predicted ratio mask, $\mathbf{M}_{*t}$ and $\mathbf{F}_{*t}$ denote their t-th frames. $\tilde{\mathbf{F}}$ is the feature map after masking. The auxiliary context embedding $\mathbf{e}$ serves as a dynamic bias vector that controls the activation threshold.

In context-aware speech enhancement [19, 22], speaker embedding or noise embedding are extracted by pre-trained embedding models. Context-aware masking (CAM), on the other hand, dynamically extracts context embedding. This method does not require extra clean utterance from the speaker of interest as reference. It automatically finds the *main speaker* in forwarding propagation.

CAM is designed to recognize a speaker as the main speaker if any of the following conditions are met: (1) The majority of speech in an utterance comes from this speaker. (2) This speaker's volume is significantly higher than others. Other speakers that appear in the utterance are called *interfering speakers*. It is beneficial to omit the speech of interfering speakers and non-speech noise in speaker verification tasks.

A simple approach to find the main speaker and global non-speech noise is to extract an utterance-level embedding. As shown in Figure 1 (c), we extract context embedding based on the input feature map of the hidden layer. First, we combine all frames with a statistics pooling layer:

$$\boldsymbol{\mu} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{F}_{*t}, \tag{9}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \mathbf{F}_{*t} \odot \mathbf{F}_{*t} - \boldsymbol{\mu} \odot \boldsymbol{\mu}}, \tag{10}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\sigma}$ is the standard deviation vector.

Following the statistics pooling layer, an FC layer maps the mean and standard deviation vectors into context embedding:

$$\mathbf{e} = \mathbf{W}_3^\top [\boldsymbol{\mu}, \boldsymbol{\sigma}] + \mathbf{b}_3, \tag{11}$$

where $[\cdot]$ denotes concatenation.

CAM can be applied multiple times to different layers. In our experiment, we apply it to the first position-wise FC layer in the vanilla TDNN and the transition layer in each block of D-TDNN. The context embedding size is half of the output size of the selected hidden layer.

Table 1. Tasks on VoxCeleb1 dataset. Here 'O' denotes 'original', 'E' denotes 'extended', and 'H' denotes 'hard'.

# of	VoxCeleb1-O	VoxCeleb1-E	VoxCeleb1-H
Speakers	40	1,251	1,251
Trial pairs	37,611	579,818	550,894

4. EXPERIMENT

4.1. Dataset

To evaluate the effectiveness of CAM, we conduct experiments on the popular speaker verification dataset VoxCeleb [24]. The utterances in VoxCeleb are collected from online videos through an automated pipeline and across unconstrained conditions. Thus, there are both clean and noisy utterances in VoxCeleb.

VoxCeleb consists of two subsets, including VoxCeleb1 and VoxCeleb2. We use the development set of VoxCeleb2 for training the models, which contains 5,994 speakers. After that, we evaluate the models on VoxCeleb1. As shown in Table 1, there are three tasks on VoxCeleb1, and the last two tasks have more trial pairs.

4.2. Implementation Details

The input features are 80-dimensional log-Mel filter banks extracted over a 25 ms long window for every 10 ms. We also adopt cepstral mean normalization over a 3-second sliding window. All data preparation steps mentioned above are processed in the Kaldi toolkit [25].

Data augmentation is also commonly used in training robust speaker embedding models. Therefore, we adopt similar strategies in [26] to augment data, including simulating reverberation with the RIR dataset [27], adding noise with the MUSAN dataset [28], and changing tempo. We also adopt SpecAugment [29], which randomly masks 0 to 10 frequency channels and 0 to 5 time frames.

We implement all models in PyTorch. We refer the reader to [5] and its GitHub page for the detailed structure of D-TDNN. We train all models with angular additive margin softmax (AAM-Softmax) loss [30]. The margin and scaling factor of AAM-Softmax loss are set to 0.25 and 32, respectively. We adopt the stochastic gradient descent (SGD) optimizer, and the mini-batch size is 128. The momentum is 0.95, and the weight decay is 5e-4. We randomly crop a 400-frame sample from each spectrogram when we construct mini-batches. The initial learning rate is 0.01, and we divide the learning rate by 10 at the 120K-th and 180K-th iterations. Training terminates at the 240K-th iterations.

We adopt cosine scoring and adaptive score normalization (AS-Norm) [31] for all models. The imposter cohort consists of the averages of the $\ell 2$-normalized speaker embeddings of each training speaker. The size of the imposter cohort is 1000.

Table 2. Results on the VoxCeleb1 dataset. The number of floating-point operations (FLOPs) is counted per sample (400 frames).

Backbone	ASP	CAM	Params (M)	FLOPs (G)	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
					EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
TDNN			4.4	1.13	1.72	0.1961	1.85	0.1918	3.06	0.2773
TDNN		✓	4.9	1.24	1.48	0.1333	1.59	0.1661	2.69	0.2382
D-TDNN			2.8	0.93	1.54	0.1938	1.65	0.1695	2.81	0.2417
D-TDNN	✓		2.9	0.96	1.50	0.1555	1.63	0.1667	2.70	0.2438
D-TDNN		✓	4.0	1.13	**1.13**	**0.1152**	**1.29**	**0.1362**	**2.31**	**0.2123**

Table 3. Results of D-TDNN in VoxCeleb1-H task.

Masking	Context-Aware	Threshold	EER (%)	MinDCF
			2.81	0.2417
✓		Fixed	2.65	0.2327
✓	✓	Dynamic	**2.31**	**0.2123**

Table 4. The number of FLOPs per sample during inference.

	D-TDNN	+ SE	+ CAM
FLOPs (G)	0.93	+ 15.32	+ 0.20

4.3. Results

We adopt two widely used metrics in speaker verification tasks, including the equal error rate (EER) and the normalized minimum detection cost function (MinDCF). The target probability is 0.01, and the costs of false alarm and miss are both 1. Table 2 shows the results on the VoxCeleb1 dataset.

Comparing the two speaker embedding backbones, we find D-TDNN outperforms the vanilla TDNN, and the average relative improvements in EER and MinDCF are 9.8% and 8.5%, respectively. When applied to the vanilla TDNN, CAM relatively decreases the EER by 14%, 14.1%, and 12.1% in the three tasks. When applied to D-TDNN, CAM relatively decreases the EER by 26.6%, 21.8%, and 17.8% in the three tasks. Similarly, CAM decreases the MinDCF by a large margin for both backbones in the three tasks.

Comparing the results on D-TDNN, we find that CAM outperforms ASP by a large margin. For example, after applying ASP and CAM, the average relative improvements in EER are 2.6% and 22.1%, respectively. One reason is that CAM assigns different mask values to different channels, which is more flexible than ASP. Another reason is that CAM is better at distinguishing signal and noise than ASP, thanks to its auxiliary context embedding.

Furthermore, we find CAM is more influential on D-TDNN than the vanilla TDNN. For example, when applying CAM to the vanilla TDNN and D-TDNN, the average relative improvements in EER are 13.4% and 22.1%, respectively. Partially because we apply CAM twice in D-TDNN. More importantly, D-TDNN is mightier than the vanilla TDNN, which helps generate more informative context embeddings and masks. This phenomenon also indicates that stronger speaker embedding backbones even benefit more from CAM.

Table 3 shows the results of masking with fixed and dynamic thresholds. Here fixed threshold means we replace the context embedding $\mathbf{e}$ with a normal bias vector $\mathbf{b}_1$. The results prove that the auxiliary context embedding is an essential component in CAM.

Table 4 shows the computational cost of speaker embedding, speech enhancement (SE) [15], and CAM. Compared to SE, CAM only requires a few percent of the computational cost.

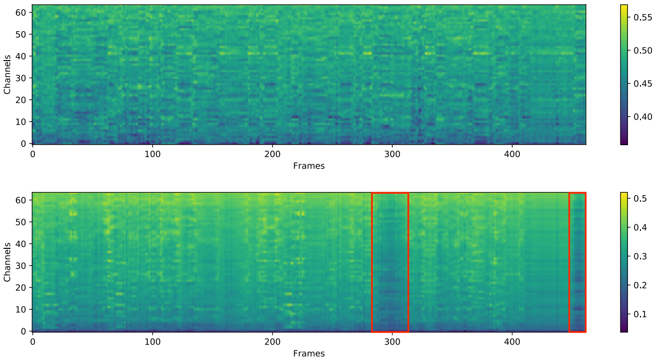

Fig. 3. Example of context-aware masks. (a) Upper: Mask in the first block of D-TDNN, applied to the shallow layer. (b) Lower: Mask in the second block of D-TDNN, applied to the deep layer. There is continuous background music in the whole utterance and an interfering speaker in the red-boxed area.

4.4. Visualization

Figure 3 is an example of context-aware masks, showing the mask values assigned to the shallow layer and the deep layer, respectively. As we can see, CAM assigns different mask values to different channels and different frames. The shaded area is considered less related to the speaker of interest. The mask applied to the deep layer forms a clear pattern of light and shaded areas. As highlighted in the red boxes, the shaded areas align to the voice of an interfering speaker. This phenomenon indicates that CAM has learned to automatically focus on the main speaker and blur interfering speakers. The mask applied to the shallow layer, on the other hand, plays a different role. Non-speech noise, such as background music, can be handled in the shallow layer at an earlier stage.

5. CONCLUSION

In this paper, we propose CAM, a novel method for robust end-to-end speaker embedding. It can learn to focus on the speaker of interest automatically. The positive results show that CAM can handle noisy data from different types of complicated scenarios. For future work, we are interested in exploring its performance on more noise-heavy data, especially those with interfering speakers. Despite the high computational cost of speech enhancement, we do not deny the effectiveness of some conventional speech enhancement methods trained in a supervised manner. It is worthy of exploring how to integrate supervised approaches to improve the performance further.

Acknowledgments This work has been supported in part by the NSFC-NRF Joint Research Project (No. 61861146001).

6. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[2] Ya-Qi Yu, Lei Fan, and Wu-Jun Li, "Ensemble additive margin softmax for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6046–6050.

[3] Lei Fan, Qing-Yuan Jiang, Ya-Qi Yu, and Wu-Jun Li, "Deep hashing for speaker identification and retrieval," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2908–2912.

[4] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, François Grondin, Réda Dehak, Leibny Paola García-Perera, Daniel Povey, Pedro A. Torres-Carrasquillo, Sanjeev Khudanpur, and Najim Dehak, "State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 1488–1492.

[5] Ya-Qi Yu and Wu-Jun Li, "Densely connected time delay neural network for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 921–925.

[6] Siqi Zheng, Yun Lei, and Hongbin Suo, "Phonetically-aware coupled network for short duration text-independent speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 926–930.

[7] Gautam Bhattacharya, Md. Jahangir Alam, Patrick Kenny, and Vishwa Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 192–198.

[8] Waad Ben Kheder, Driss Matrouf, Pierre-Michel Bousquet, Jean-François Bonastre, and Moez Ajili, "Fast i-vector denoising using MAP estimation and a noise distributions database for robust speaker recognition," *Computer Speech and Language*, vol. 45, pp. 104–122, 2017.

[9] Mohammad MohammadAmini, Driss Matrouf, and Paul-Gauthier Noé, "Denoising x-vectors for robust speaker recognition," in *Speaker and Language Recognition Workshop (Odyssey)*, 2020, pp. 75–80.

[10] DeLiang Wang and Jitong Chen, "Supervised speech separation based on deep learning: An overview," *IEEE ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 26, no. 10, pp. 1702–1726, 2018.

[11] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "SEGAN: speech enhancement generative adversarial network," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 3642–3646.

[12] Daniel Michelsanti and Zheng-Hua Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2008–2012.

[13] Phani Sankar Nidadavolu, Saurabh Kataria, Jesús Villalba, L. Paola García-Perera, and Najim Dehak, "Unsupervised feature enhancement for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7599–7603.

[14] Saurabh Kataria, Phani Sankar Nidadavolu, Jesús Villalba, Nanxin Chen, L. Paola García-Perera, and Najim Dehak, "Feature enhancement with deep feature losses for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7584–7588.

[15] Suwon Shon, Hao Tang, and James R. Glass, "VoiceID loss: Speech enhancement for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2888–2892.

[16] Youngmoon Jung, Yeunju Choi, Hyungjun Lim, and Hoirin Kim, "A unified deep learning framework for short-duration speaker verification in adverse environments," *IEEE Access*, vol. 8, pp. 175448–175466, 2020.

[17] Yanpei Shi, Qiang Huang, and Thomas Hain, "Robust speaker recognition using speech enhancement and attention model," *CoRR*, vol. abs/2001.05031, 2020.

[18] Fu-Kai Chuang, Syu-Siang Wang, Jeih-weih Hung, Yu Tsao, and Shih-Hau Fang, "Speaker-aware deep denoising autoencoder with embedded speaker identity for speech enhancement," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 3173–3177.

[19] Quan Wang, Hannah Muckenhirn, Kevin W. Wilson, Prashant Sridhar, Zelin Wu, John R. Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez-Moreno, "VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2728–2732.

[20] Seongkyu Mun, Soyeon Choe, Jaesung Huh, and Joon Son Chung, "The sound of my voice: Speaker representation loss for target voice separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7289–7293.

[21] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014, pp. 2670–2674.

[22] Joohyung Lee, Youngmoon Jung, Myunghun Jung, and Hoirin Kim, "Dynamic noise embedding: Noise aware training and adaptation for speech enhancement," *CoRR*, vol. abs/2008.11920, 2020.

[23] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2252–2256.

[24] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, 2020.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[26] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 3830–3834.

[27] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[28] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A music, speech, and noise corpus," *CoRR*, vol. abs/1510.08484, 2015.

[29] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2613–2617.

[30] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[31] Sandro Cumani, Pier Domenico Batzu, Daniele Colibro, Claudio Vair, Pietro Laface, and Vasileios Vasilakakis, "Comparison of speaker recognition approaches for real applications," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 2365–2368.