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

Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System

Zhifu Gao[1], *Yan Song*[1], *Ian McLoughlin*[2], *Pengcheng Li*[1], *Yiheng Jiang*[1], *Lirong Dai*[1]

[1]University of Science and Technology of China, Hefei, China
[2]School of Computing, University of Kent, Medway, UK
{gaozf, pclee, jiangyh}@mail.ustc.edu.cn, {songy, lrdai}@ustc.edu.cn, ivm@kent.ac.uk

Abstract

Deep embedding learning based speaker verification (SV) methods have recently achieved significant performance improvement over traditional i-vector systems, especially for short duration utterances. Embedding learning commonly consists of three components: frame-level feature processing, utterance-level embedding learning, and loss function to discriminate between speakers. For the learned embeddings, a back-end model (*i.e.*, Linear Discriminant Analysis followed by Probabilistic Linear Discriminant Analysis (LDA-PLDA)) is generally applied as a similarity measure. In this paper, we propose to further improve the effectiveness of deep embedding learning methods in the following components: (1) A multi-stage aggregation strategy, exploited to hierarchically fuse time-frequency context information for effective frame-level feature processing. (2) A discriminant analysis loss is designed for end-to-end training, which aims to explicitly learn the discriminative embeddings, *i.e.* with small intra-speaker and large inter-speaker variances. To evaluate the effectiveness of the proposed improvements, we conduct extensive experiments on the VoxCeleb1 dataset. The results outperform state-of-the-art systems by a significant margin. It is also worth noting that the results are obtained using a simple cosine metric instead of the more complex LDA-PLDA backend scoring.

Index Terms: speaker verification, speaker embedding, multi-stage aggregation, discriminant analysis loss

1. Introduction

Speaker recognition (SR) is the task of automatically retrieving identity information from a given speech utterance. Generally, it can be categorized into speaker identification (SID) and speaker verification (SV), according to the recognition settings. The former classifies an utterance into a specific identity from a known speaker set, while the latter determines whether the claimed identity of a speaker matches a given enrolment.

Compared to SID, SV is an open-set recognition problem, which means there is no overlap in speakers between the training and test set. In essence, this means that SV is closely related to the metric learning problem, where the key is to learn effective utterance-level representations with small intra-class and large inter-class variances.

In recent years, more attention has been paid to deep learning methods for SV, where deep neural networks (DNNs) are employed to extract speaker representations. Being able to benefit from a discriminative training process, deep embedding methods such as d-vector or x-vector have been shown to outperform traditional i-vectors [1, 2], especially for short duration utterances. Existing deep embedding learning architectures include time-delay DNN (TDNN) [2], convolutional neural network (CNN) [3, 4], and Long Short-Term Memory Network (LSTM) [5]. They generally consist of three main components [6, 7]: (1) Frame-level feature processing to model local short spans of acoustic features via TDNN or convolutional layers. (2) Utterance-level embedding learning, containing a pooling method to map variable-length frame-level features into fixed-length utterance-level representations.(3) Loss function to discriminate directly between speakers. An LDA-PLDA backend has proved to be crucial to improve performance of i-vector based SV systems, since it can effectively compensate for channel differences [8, 9, 10, 11]. It is also widely used in deep embedding based systems as a back-end model [2, 4, 12].

Many recent works have focused on utterance-level embedding learning, $e.g.$, average pooling [1], statistical pooling [2], attentive pooling [13, 14], cross-convolutional-layer pooling [3], learnable dictionary encoding (LDE) [12]. Besides cross entropy loss (CE), different loss functions have been recently proposed, including triplet loss [15, 16], center loss [12, 17], angular softmax (A-softmax) [12, 18], additive margin softmax (AM-softmax) [19] and logistic margin (LM) [19]. However, it is still challenging to incorporate the effective LDA-PLDA backend into a deep embedding learning architecture. Furthermore, few works have considered frame-level processing [7]. In this paper, we focus on frame-level processing and the loss function for more effective embedding learning while, for utterance-level learning, we employ a statistical pooling method [2]. The overall framework is shown in Fig.1, and will be described in more detail in Section 2.

For frame-level processing, a multi-stage aggregation[1] (MSA) strategy is proposed to exploit hierarchical time-frequency context information. Each stage contains a sequence of convolutional layers, and outputs the feature maps with different channels and time-frequency resolutions. In MSA, the outputs of stages are first convoluted to match time-frequency resolutions, then incorporated into embeddings. This differs from existing frame-level processing, in which the learned features are generally over a single scale.

In terms of loss function, a discriminant analysis loss (DALoss) is proposed to overcome the shortcoming that LDA-PLDA could not be jointly trained with the embedding learning network. This is motivated by work in computer vision [20, 21]. The LDA-PLDA backend cannot be jointly trained with the embedding learning model in current SV systems, but our proposed DALoss can work end-to-end, allowing LDA-PLDA to be jointly trained with CE loss.

To evaluate the effectiveness of the proposed MSA and DALoss, extensive experiments have been conducted. Results show the proposed method obtains 17% relative improvement in terms of equal error rate (EER) over the baseline. To our

[1]Here, the aggregation is defined as the combination of different stages throughout the network instead of the pooling operation itself.

http://dx.doi.org/10.21437/Interspeech.2019-1489

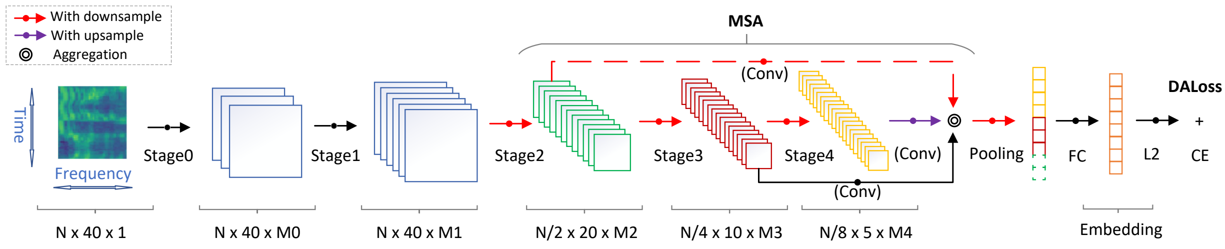

Figure 1: *Illustration of network architecture. We use different colours to indicate different resolutions in the time-frequency domain, e.g., N×40×1 (corresponding to time, frequency and channel axis respectively), when colour changes (stage2⋯4), resolution halves(in both time and frequency axis).*

knowledge, this also outperforms other state-of-the-art systems, as detailed in Sec. 4.2.

2. Overview of network architecture

The proposed end-to-end deep embedding learning architecture is shown in Fig. 1. It consists of frame-level processing, utterance-level embedding learning, and loss functions.

The frame-level processing part could be divided into 5 stages according to their time-frequency resolutions, as annotated in the figure, where each stage consists of a sequence of convolutional layers. Stages 0 and 1 have the same time-frequency resolution (*e.g.* $N\times40$). The resolution of feature maps then halves in both time and frequency axes from stage 2 to 4, generating hierarchical features with pyramidal scales. Viewing Fig. 1 from left to right, resolutions change from fine to coarse, corresponding to hierarchical feature context information from local to global [22]. To effectively utilize the hierarchical features from different stages for frame-level feature processing, an MSA strategy is designed which consists of a convolutional layer at each stage to match the time-frequency resolution, and a concatenation operation to fuse them.

A pooling layer then follows the frame-level part to map frame-level features into utterance-level representations. In this paper, we use statistical pooling [2], although it is easy to extend to the other pooling methods mentioned in Section 1. A fully connected (FC) layer, termed an embedding layer, is inserted to make a nonlinear transformation of speaker representations.

The outputs of the FC are firstly length normalized by L2 and then multiplied by a constant scale [4], before being fed into CE and DALoss. To overcome the shortcoming that LDA-PLDA could not be jointly trained with the embedding learning network, we have designed DALoss to perform like LDA-PLDA in an end-to-end network, which could be jointly trained with CE loss. The DALoss is expected to imitate the LDA-PLDA backend in two ways. On one hand, it reduces intra-speaker variances, to make embeddings for each speaker identity more compact. On the other hand, it enlarges inter-speaker variances, allowing the network to focus on differences caused by speaker identity rather than variability caused by different channels, gender or speech content.

After the network is trained, speaker embeddings are extracted from the embedding layer for the enrolment and test set. Finally, cosine similarities are calculated between embeddings to perform speaker verification.

3. Methods

3.1. Multi-stage aggregation (MSA)

Although it is feasible to aggregate any stages throughout the network, we are concerned with aggregating the outputs of stages with different resolutions. We design an MSA strategy to utilize multiple inputs with different resolutions and incorporate them into the outputs, as depicted in Fig. 1.

Specially, we aggregate the outputs from stage 2 to stage 4. We note feature maps of the outputs of the l-th stage as, $[\boldsymbol{x}_1^l, \boldsymbol{x}_2^l, ..., \boldsymbol{x}_{M_l}^l], l = 2, 3, 4$, where M_l denotes the number of channels. Since they have different feature map sizes, we utilize convolutions to make them match in time-frequency resolution, as formulated as:

$$[\widehat{\boldsymbol{x}}_1^l, \widehat{\boldsymbol{x}}_2^l, ..., \widehat{\boldsymbol{x}}_{M_l}^l] = Conv(\boldsymbol{x}_1^l, \boldsymbol{x}_2^l, ..., \boldsymbol{x}_{M_l}^l) \tag{1}$$

where $Conv$ could be a single convolution with stride of 2 to downsample the feature maps. It could also be a single transposed convolution or bilinear interpolation to upsample. They are then concatenated and fed into a nonlinear transition layer, formulated as:

$$\Gamma(\boldsymbol{X}) = H_\Gamma([\widehat{\boldsymbol{x}}_1^2, ..., \widehat{\boldsymbol{x}}_{M_2}^2; \widehat{\boldsymbol{x}}_1^3, ..., \widehat{\boldsymbol{x}}_{M_3}^3; \widehat{\boldsymbol{x}}_1^4, ..., \widehat{\boldsymbol{x}}_{M_4}^4]) \tag{2}$$

Though the transition H_Γ can be based on any layers, for efficiency and simplicity, we choose a single convolution with kernel size of 1x1 and stride of 1 followed by batch normalization [25] and a nonlinearity.

3.2. Discriminant analysis loss (DALoss)

In this subsection, we formulate the DALoss mathematically. We suppose that there are K identities and each identity comprises N utterances in a min-batch. $\boldsymbol{x}_n^k$ denotes the n-th embedding of the k-th identity. DALoss could be formulated as:

$$L_{DALoss} = \beta S_{intra} + \gamma S_{inter} \tag{3}$$

where S_{intra} denotes the intra-speaker variabilities and S_{inter} represents the inter-speaker variabilities. β and γ are their loss weights of each respective variance.

Since S_{intra} penalizes the maximum variabilities within each speaker, it could be formulated as:

$$S_{intra} = \sum_{k=1}^{K} S_{intra}^k = \sum_{k=1}^{K} \frac{C^k}{\sum_{j=1}^{C^k} \frac{1}{f_j(\boldsymbol{x}_m^k, \boldsymbol{x}_n^k)}} \tag{4}$$

Table 1: *Performance of state-of-the-art systems on VoxCeleb1*

Frame-level model	Pooling method	Loss funtion	Similarity	EER (%)
I-vectors [23]	-	-	PLDA	8.8
X-vectors [24]	Statistical pooling	CE	PLDA	7.1
VGG-M [23]	Average pooling	Contrastive	Cosine	7.8
ResNet-34 [12]	LDE	Center loss	PLDA	4.87
		A-softmax		4.48
ResNet-20 [19]	Average pooling	LM	Cosine	4.42
		A-softmax		4.3
		AM-softmax		4.29
R-MSA_3_4	Statistical pooling	DALoss-E	Cosine	4.12
		DALoss-C		**4.09**
X-MSA_3_4	Statistical pooling	DALoss-E	Cosine	3.95
		DALoss-C		**3.87**

where $f_j(\boldsymbol{x}_m^k, \boldsymbol{x}_n^k)$ denotes the j-th largest distance between embeddings of the k-th identity. The overall cost is the mean of the first C^k-th largest distances within each identity. The S_{intra} loss is designed to compress the distance of those hard samples of the same identity and thus reduce the largest intra-speaker variabilities.

Then S_{inter} represents the minimum variabilities between speakers, formulated as:

$$S_{inter} = \max(0, m - \min(f(\widetilde{\boldsymbol{x}}^i, \widetilde{\boldsymbol{x}}^j))) \tag{5}$$

$$\widetilde{\boldsymbol{x}}^l = \frac{1}{N^l} \sum_{n=1}^{N^l} \boldsymbol{x}_n^l \tag{6}$$

where $\widetilde{\boldsymbol{x}}^l$ denotes the center of the l-th identity in current min-batch. m denotes a super parameter of the minimum margin between identity centers. N^l is the number of embedding of the l-th identity. $i \neq j \in [1, ..., K]$. The model pulls the distances between centers of different identities to be larger than the minimum margin by reducing S_{inter} loss.

We define two kinds of distance metric, formulated as:

$$f_E(\boldsymbol{x}, \boldsymbol{y}) = \parallel \boldsymbol{x} - \boldsymbol{y} \parallel_2^2 \tag{7}$$

$$f_C(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\parallel \boldsymbol{x} \parallel_2^2 \parallel \boldsymbol{y} \parallel_2^2} \tag{8}$$

where $f_E(\boldsymbol{x}, \boldsymbol{y})$ denotes Euclidean distance and $f_C(\boldsymbol{x}, \boldsymbol{y})$ denotes cosine distance, termed *DALoss-E* and *DALoss-C* respectively in the following experiments.

4. Experiments

4.1. Experiment setup

We evaluate performance on Voxceleb1 without data augmentation. The audio is converted to 41-dimensional filter bank outputs (FBank), with a frame-length of 25 ms, and mean-normalized over a sliding window of 3 s. These FBank features are randomly truncated into short slices ranging from 2 s to 4 s, finally generating 3200 input slices per speaker. The network optimizer is stochastic gradient descent (SGD) with a momentum rate of 0.9. The GPU platform is a single GTX1080Ti card and, limited by the GPU memory, we set $K = 50, C^k = 2, N = 2$ in a min-batch. The learning rate is initialized to 0.1, and multiplied by 0.1 every epoch. β, γ is set to 0.1 and $m = 0.2$ in this paper.

To evaluate the effectiveness of MSA and DALoss, We utilize ResNet [26] and ResNetXt [27] as the network backbone respectively, detailed in Table 2. In this case, stage 0 consists of a single convolution layer with kernel size of 7x7, stride of 1 and padding of 3. Batch normalization and ReLu are added. In the following experiments, when the system name begins with the "**R**" or "**X**", it means the backbone was ResNet or ResNetXt respectively.

Table 2: *Backbones of the network*

—	ResNet		ResNetXt	
Layer	Channels	Blocks	Channels	Blocks
Stage 0	16	-	16	-
Stage 1	16	3	16	3
Stage 2	32	4	32	3
Stage 3	64	6	64	4
Stage 4	128	3	128	3
Embedding	512			

4.2. Comparison with state-of-the-art systems

We have compared our proposed systems with current state-of-the-art systems, as shown in Table 1. All these systems were trained on VoxCeleb1 [23], without using data augmentation.

R-MSA_3_4 and *X-MSA_3_4* are our proposed systems, which are combined with MSA and DALoss. In both of them, the MSA module aggregates the outputs of stage 3 and stage 4. The *DALoss-E* and *DALoss-C* in the loss column, denotes that the DALoss distance is measured by Euclidean or cosine distance respectively.

Results reveal that the performance of *DALoss-C* is sightly superior to *DALoss-E*. It may because there exists a mismatch between training and evaluation criterions in *DALoss-E*. Under the DALoss-C, *R-MSA_3_4* and *X-MSA_3_4* obtain 11% and 17% relative improvements over baseline *R* and *X* without MSA and DALoss respectively, described in Section 4.3.

The i-vector and x-vector systems are two widely used baselines. Cai *et al.* [12] investigated the LDE pooling method and discriminative loss, *e.g.*, center loss and A-softmax. LM was proposed for building an end-to-end system in [19]. These systems all obtained large improvements in performance over the baselines. Our proposed systems which incorporate MSA and DALoss have all achieved performance comparable with state-of-the-art. It is worth noting that our systems adopt a simple cosine metric as a similarity measure.

4.3. Evaluation of MSA

In this subsection, we evaluate the performance of MSA separately, described in Table 3. The LDA-PLDA backend is applied to calculate similarity scores, and we do not add DALoss in this subsection. "DCF" in Table 3, denotes the minimum of the normalized detection cost function at P_{Target}=0.01($minDCF_{0.01}$). The configurations for comparison are as follows:

R: This is a baseline without MSA, where the backbone network is ResNet-34 [26], as detailed in Table 2.

R-MSA_3_4/_2_3_4: These use our proposed MSA method, where we adopt *ResNet* as backbone. In *R-MSA_3_4*, the MSA module aggregates the outputs of stage 3 and stage 4. In *R-MSA_2_3_4*, the MSA module aggregates the outputs of stage 2, stage 3 and stage 4.

X: This is another baseline without MSA, where the backbone network is ResNetXt-41 [27], as detailed in Table 2.

X-MSA_3_4/_2_3_4: These also use our proposed MSA method, where *ResNetXt* is adopted as backbone. In *X-MSA_3_4*, the MSA module aggregates the outputs of stage 3 and stage 4. In *X-MSA_2_3_4*, the MSA module aggregates the outputs of stage 2, stage 3 and stage 4.

Table 3: *Results of MSA (EER% / $minDCF_{0.01}$)*

System	Cosine		LDA-PLDA		Size
	EER	DCF	EER	DCF	-
R	4.62	**0.4643**	4.51	0.4830	2M
R-MSA_3_4	**4.41**	0.4645	**4.25**	**0.4587**	2.1M
R-MSA_2_3_4	4.53	0.4681	4.43	0.4593	2.3M
X	4.69	0.4397	4.11	0.4418	1.2M
X-MSA_3_4	**4.31**	**0.3956**	**3.85**	**0.4257**	1.4M
X-MSA_2_3_4	4.31	0.4058	3.95	0.4425	1.8M

Firstly, we compare the systems whose backbone networks use *ResNet*. Compared to *R*, *R-MSA_3_4* obtains a stable improvement, yielding 6%/5% relative improvements in EER and minDCF respectively (for LDA-PLDA). It aggregates frame-level features from stage 3 and stage 4 with only a small increase in the number of parameters (shown in the Size column). When we further aggregate features from stage 2, stage 3 and stage 4, the performance declines slightly compared with *R-MSA_3_4*. We infer the reason may be that features from stage 2 are weak in context information and thus do not contribute efficiently to discriminative embedding. When they are aggregated into deep embedding by MSA, the network actually tends to become shallower, because part of the information flow skips stages 3 and stage 4.

Looking now at the systems whose backbone network uses *ResNetXt*, the results are similar to those of the *ResNet* system variants. Specially, *X-MSA_3_4* obtains a 9% relative improvement over *X* in both EER and minDCF respectively (for LDA-PLDA). As with the *ResNet* backbone systems, *X-MSA_2_3_4* decays, too, tending to confirm our supposition above.

Comparing the two backbone system variants, the results show that performance of systems with MSA can improve upon baseline systems without MSA. It empirically convinces us that features with different resolutions are complementary; fine-resolution features from lower stages are rich in local short representation while coarse-resolution features from higher stages contain global context information. The proposed MSA strategy is an effective method to fuse these complementary items of information.

4.4. Evaluation of DALoss

In this subsection, we evaluate the performance of DALoss separately, as described in Table 4. For the sake of clarity, we only show the performance of *R-MSA_3_4* described in subsection 4.3, since the DALoss is independent of specific embedding learning networks. DALoss is jointly trained with CE loss and we omit CE for short in Table 4.

Table 4: *Results of DALoss (EER% / $minDCF_{0.01}$)*

System	Loss	Metric	EER	DCF
R-MSA_3_4	CE	Cosine	4.41	0.4656
		PLDA	4.25	0.4587
	DALoss-E	Cosine	4.12	**0.4570**
	DALoss-C		**4.09**	0.4578

From Table 4, we can see that the DALoss significantly improves the Cosine performance in both *DALoss-E* (Euclidean distance) and *DALoss-C* (cosine distance). Specially, the *DALoss-C* obtains 7% relative improvements over the CE loss in EER (Cosine). In fact, when *R-MSA_3_4* is trained with DALoss-E/DALoss-C, its cosine performance outperforms the LDA-PLDA backend. We think the gains come from the joint training, since the LDA-PLDA cannot be jointly trained with the embedding learning network.

From the above comparisons, we could conclude that the DALoss method is able to improve performance, and even outperform the LDA-PLDA backend. The results confirm that DALoss can effectively reduce intra-speaker variances and enlarge inter-speaker variances, leading to the ability to generate more discriminative speaker embeddings.

5. Conclusions

This paper has proposed an improved end-to-end deep embedding learning system for SV. To further enhance performance of deep embedding learning methods, we have proposed MSA and DALoss. The MSA is exploited to incorporate hierarchical features of pyramidal scales and resolutions into speaker embeddings. The results show that MSA is an effective strategy to fuse these multiple complementary features. We have also proposed DALoss to learn more discriminative embeddings with smaller intra-speaker variances and larger inter-speaker variances. It is inherently built in an end-to-end fashion to create a network which can be jointly trained using CE loss, unlike that using LDA-PLDA. Extensive experiments have been conducted on VoxCeleb1. The results show the proposed method obtains 17% relative improvement over baseline to achieve state-of-the-art performance.

Since there are some hyper-parameters in DALoss, it may need some skills to set their values for different conditions. In future, we would extend DALoss to other conditions to study the dependencies between the setups of hyper-parameters and datasets. Moreover, we hope to reduce the number of hyper-parameters.

6. Acknowledgements

The work was supported by National Natural Science Foundation of China grant no U1613211.

7. References

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

[2] 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.

[3] Z. Gao, Y. Song, I. McLoughlin, W. Guo, and L. Dai, “An improved deep embedding learning method for short duration speaker verification,” *Proc. Interspeech 2018*, pp. 3578–3582, 2018.

[4] W. Cai, J. Chen, and M. Li, “Analysis of length normalization in end-to-end speaker verification system,” *Proc. Interspeech 2018*, pp. 3618–3622, 2018.

[5] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[6] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, “Utterance-level aggregation for speaker recognition in the wild,” *arXiv preprint arXiv:1902.10107*, 2019.

[7] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, “Deep speaker embedding learning with multi-level pooling for text-independent speaker verification,” *arXiv preprint arXiv:1902.07821*, 2019.

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

[9] 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, 2011.

[10] 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.

[11] 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, 2015.

[12] W. Cai, J. Chen, and M. Li, “Exploring the encoding layer and loss function in end-to-end speaker and language recognition system,” *arXiv preprint arXiv:1804.05160*, 2018.

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

[14] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” *Proc. Interspeech 2018*, pp. 3573–3577, 2018.

[15] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: an end-to-end neural speaker embedding system,” *arXiv preprint arXiv:1705.02304*, 2017.

[16] S. Novoselov, V. Shchemelinin, A. Shulipa, A. Kozlov, and I. Kremnev, “Triplet loss based cosine similarity metric learning for text-independent speaker recognition,” *Proc. Interspeech 2018*, pp. 2242–2246, 2018.

[17] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, “Deep discriminative embeddings for duration robust speaker verification,” *Proc. Interspeech 2018*, pp. 2262–2266, 2018.

[18] Z. Huang, S. Wang, and K. Yu, “Angular softmax for short-duration text-independent speaker verification,” *Proc. Interspeech, Hyderabad*, 2018.

[19] M. Hajibabaei and D. Dai, “Unified hypersphere embedding for speaker recognition,” *arXiv preprint arXiv:1807.08312*, 2018.

[20] M. Dorfer, R. Kelz, and G. Widmer, “Deep linear discriminant analysis,” *arXiv preprint arXiv:1511.04707*, 2015.

[21] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, “Range loss for deep face recognition with long-tailed training data,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5409–5418.

[22] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, “Deep layer aggregation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 2403–2412.

[23] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[24] S. Shon, H. Tang, and J. Glass, “Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model,” *arXiv preprint arXiv:1809.04437*, 2018.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.