INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

An Effective Speaker Recognition Method Based on Joint Identification and Verification Supervisions

Ying Liu[1], *Yan Song*[1], *Yiheng Jiang*[1], *Ian McLoughlin*[1,2], *Lin Liu*[3], *Lirong Dai*[1]

[1]National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei, China.
[2]ICT Cluster, Singapore Institute of Technology
[3]iFLYTEK Research, iFLYTEK CO., LTD., Hefei, Anhui 230088, China.

{ly1004, jiangyh}@mail.ustc.edu.cn, {songy, ivm, lrdai}@ustc.edu.cn, linliu@iflytek.com

Abstract

Deep embedding learning based speaker verification methods have attracted significant recent research interest due to their superior performance. Existing methods mainly focus on designing frame-level feature extraction structures, utterance-level aggregation methods and loss functions to learn discriminative speaker embeddings. The scores of verification trials are then computed using cosine distance or Probabilistic Linear Discriminative Analysis (PLDA) classifiers. This paper proposes an effective speaker recognition method which is based on joint identification and verification supervisions, inspired by multi-task learning frameworks. Specifically, a deep architecture with convolutional feature extractor, attentive pooling and two classifier branches is presented. The first, an identification branch, is trained with additive margin softmax loss (AM-Softmax) to classify the speaker identities. The second, a verification branch, trains a discriminator with binary cross entropy loss (BCE) to optimize a new triplet-based mutual information. To balance the two losses during different training stages, a ramp-up/ramp-down weighting scheme is employed. Furthermore, an attentive bilinear pooling method is proposed to improve the effectiveness of embeddings. Extensive experiments have been conducted on VoxCeleb1 to evaluate the proposed method, demonstrating results that relatively reduce the equal error rate (EER) by 22% compared to the baseline system using identification supervision only.

Index Terms: speaker verification, mutual information learning, attentive bilinear pooling, multi-task framework

1. Introduction

Speaker recognition (SR) is the task of automatically determining whether a given utterance belongs to a certain speaker identity. According to different recognition settings, SR can be categorized into either speaker identification (SID) which classifies a given utterance as being from a specific speaker, or speaker verification (SV), which is a binary classification problem that determines whether two given speech utterances belong to same speaker or not. Compared to SID, SV is an open-set recognition task with no overlap between training and test set, which is closely related to representation learning.

Over the past few decades, the most popular SV methods have been based on i-vector followed by Probabilistic Linear Discriminative Analysis (PLDA) [1, 2], in which the i-vector representation is generally learned in an unsupervised manner. Recently, deep embedding learning based SV methods have attracted significant interest due to their superior performance. Compared to traditional i-vector systems, deep learning based SV methods may benefit from the discriminative characteristics and large receptive-field of deep neural networks (DNNs).

Existing deep embedding learning architectures include time-delay DNN (TDNN) [3], convolutional neural network (CNN) [4, 5, 6], and Long Short-Term Memory (LSTM) networks [7]. Generally, these architectures consist of a frame-level feature extractor, an utterance-level aggregator and a classifier, which can be optimized in an end-to-end way.

Many recent works have focused on utterance-level aggregation methods, *e.g.*, statistical pooling [3], attentive pooling [8, 9], bilinear pooling [4], and dictionary based pooling methods [10, 11]. Meanwhile, other works have proposed different loss functions, including triplet loss [12, 13], center loss [10], triplet-center loss [14], angular softmax loss (A-Softmax) [10] and additive margin softmax loss (AM-Softmax) [15, 16]. However, in most deep embedding learning methods, the network architectures are trained under identification supervision, optimized for the SID task. Meanwhile, for SV tasks, the verification score between utterance pairs is computed via cosine distance or through an additional trainable backend. It is still difficult to directly incorporate an effective backend into a deep embedding learning architecture [4].

In this paper, an effective speaker recognition method is proposed based on joint identification and verification supervisions. This is inspired by the multi-task learning framework, as shown in Fig. 1 and detailed in Section 3. Specifically, this includes a deep architecture with frame level feature extractor, attentive pooling and two branches of classifiers. The first branch is similar to deep embedding learning, in which a speaker classifier is optimized via AM-Softmax loss to discriminate the learned speaker embeddings. The second branch optimizes a new triplet-based mutual information (T-MI) between positive and negative samples extracted from the embedding space, inspired by triplet loss [12, 13] and mutual information based representation learning [17]. As with generative adversarial networks (GANs), we train a discriminator to separate them, using binary cross entropy loss (BCE). To prevent the issue of an imbalance between AM-softmax and BCE loss at different training stages, we introduce a ramp-up/ramp-down weighting scheme. In addition, a new attentive bilinear pooling method (ABP) is proposed, aiming to improve performance by aggregating features along the temporal axis.

To evaluate the effectiveness of the proposed method, extensive experiments have been conducted on the Voxceleb1 benchmark [18]. By jointly optimizing the identification and verification, our method can relatively reduce the equal error rate (EER) by 22% compared to the baseline system using identification supervision only.

Copyright © 2020 ISCA

http://dx.doi.org/10.21437/Interspeech.2020-1922

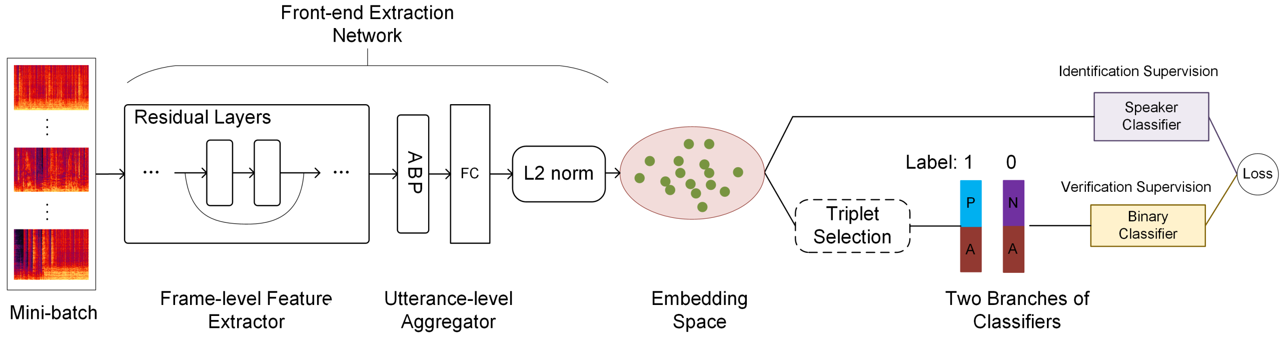

Figure 1: *Framework of the proposed SV method based on joint identification and verification supervisions.*

2. Overview of the proposed multi-task learning framework

The proposed multi-task learning based speaker recognition framework is shown in Fig. 1. It consists of frame-level feature extractor, utterance-level aggregator, and two branches of classifiers.

The frame-level feature extractor is adapted from the existing ResNet-18 architecture [19], which comprises an input convolutional layer and 4 residual stages. The main difference lies in that we keep the temporal and frequency dimensions of feature maps in each residual stage unchanged, and insert a transition layer to reduce the frequency dimension.

The aggregator is then followed to map the extracted frame-level features into utterance-level representations. In this paper, a novel attentive bilinear pooling (ABP) method is introduced to improve the effectiveness of embeddings, detailed in Section 3.3. Then an embedding layer, implemented by a fully connected (FC) layer, is added to make a nonlinear transformation and dimension reduction to obtain speaker embeddings.

The speaker embeddings are firstly length normalized and then fed into two branches of classifiers for multi-task learning. The first, an identification branch, is implemented by a FC layer and trained with AM-Softmax loss for SID task. The second, a verification branch, accomplishes the SV task by first constructing the positive and negative pairs from the selected triplet, and then training a binary classifier with BCE loss. Finally, a ramp-up/ramp-down weighting scheme is employed to balance the AM-softmax and BCE loss for multi-task learning.

During testing, we can either extract speaker embeddings from the embedding layer for the enrolment and test set, and then use a traditional PLDA backend to calculate verification scores, or directly use the output of the verification branch, giving scores in an end-to-end fashion.

3. Methods

3.1. Triplet-based mutual information (T-MI) learning

Mutual Information (MI) of statistical dependence is a promising tool for learning representations in an unsupervised way [17]. Given two random variables x and y, MI can be defined as

$$\begin{aligned} MI(x;y) &= \int_x \int_y p(x,y) \log \left\{ \frac{p(x,y)}{p(x)p(y)} \right\} dxdy \\ &= D_{KL}\{p(x,y) \| p(x)p(y)\} \end{aligned} \tag{1}$$

where D_{KL} is the Kullback-Leibler (KL) divergence between the joint distribution $p(x, y)$ and product of marginals $p(x)p(y)$. The MI is minimized when the random variable x and y are statistically independent, and is maximized when they contain identical information. For SV task, inspired by triplet loss [13], we can construct triplet (x_a, x_p, x_n), where x_a and x_p are utterances from the same speaker, x_a and x_n are from the different speakers. And then discriminative speaker representations can be effectively learned by maximizing MI between x_a and x_p, and minimizing it between x_a and x_n. This is logical, however, MI is hard to measure directly.

Fortunately, previous research [20] has found it possible to optimize the MI within an encoder-discriminator framework. Motivated by [17], a verification branch is designed as the discriminator using T-MI learning. Specifically, the front-end extraction network, including a frame-level feature extractor and an utterance-level aggregator, is used as the encoder, denoted by $f_\theta(\cdot)$. Embeddings of the triplet can be obtained by feeding it through the network. Then positive embedding pair $(f_\theta(x_a), f_\theta(x_p))$ and negative embedding pair $(f_\theta(x_a), f_\theta(x_n))$ are formed and passed through the verification branch, implemented by a binary classifier denoted by $g_\phi(\cdot)$, for discriminating verification. The positive pair and the negative pair are labeled '1' and '0' respectively, and the standard binary cross entropy loss (BCE) is used as the objective function to train the classifier:

$$\begin{aligned} L_{ver} = \frac{1}{N} \sum_{i=1}^{N} &- \log \left\{ g_\phi(f_\theta(x_a^i) \oplus f_\theta(x_p^i)) \right\} \\ &- \log \left\{ 1 - g_\phi(f_\theta(x_a^i) \oplus f_\theta(x_n^i)) \right\} \end{aligned} \tag{2}$$

where $\oplus$ denotes the concatenation operator. The BCE loss in Eq. (2) actually estimates the *Jansen-Shannon* divergence between positive and negative distributions, which is similar to the KL-based definition of MI [17].

The main difference to [17] is that we construct triplet with respect to label information in a mini-batch, which in fact introduces the verification supervision. Therefore, the output sigmoid probability of the verification branch can be used as a similarity measure of two embeddings, which conveniently allows the system to be trained end-to-end without PLDA backend or cosine distance calculation. Given input pair (x_1, x_2), the verification score can be obtained as:

$$score(x_1, x_2) = g_\phi(f_\theta(x_1) \oplus f_\theta(x_2)) \tag{3}$$

3.2. Joint optimization of identification and verification

As discussed above, the multi-task system is optimized jointly with identification and verification supervisions. For SID task, AM-Softmax loss is used as the objective function:

$$L_{iden} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s\cdot(cos\theta_{y_i}-m)}}{e^{s\cdot(cos\theta_{y_i}-m)} + \sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \tag{4}$$

where s is the scale parameter and m is the margin. In our experiments, we set s = 18 and m = 0.1.

The total loss for joint optimization is the weighted sum of the identification loss and verification loss:

$$L = \lambda L_{iden} + \mu L_{ver} \tag{5}$$

where the λ and μ are weight parameters. To balance these two loss components at different training stages, a ramp-up/ramp-down weighting scheme is introduced. The weight μ starts from zero and ramps up along the curve $\mu(t) = \mu_0 e^{-5\{1-t/T_1\}^2}$, and similarly, λ ramps down according to the curve $\lambda(t) = \lambda_0 e^{-5\{(t-T_2)/(T_3-T_2)\}^2}$, where t is training epoch, μ_0 is the final value of μ, λ_0 is the initial value of λ, [0, T_1] and [T_2, T_3] are the durations of ramp-up and ramp-down periods respectively. In our experiments, μ_0 and λ_0 are set to 1.

3.3. Attentive bilinear pooling (ABP)

Inspired by [4, 8], an attentive bilinear pooling (ABP) method is further utilized to force the model to pay more attention to useful information for aggregation. It calibrates the output frame-level features with learnable convolutional layer to provide frame-wise attention mechanism.

Specifically, let $\mathbf{H} \in \mathbf{R}^{L\times D}$ be the frame-level feature map captured by the hidden layer below the self-attention layer, where L and D are the number of frames and feature dimension respectively. Then the attention map $\mathbf{A} \in \mathbf{R}^{L\times K}$ can be obtained by feeding $\mathbf{H}$ into a 1×1 convolutional layer followed by softmax non-linear activation, where K is the number of attention heads. The 1^{st}-order and 2^{nd}-order attentive statistics of $\mathbf{H}$, denoted by $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$, can be computed similar as cross-layer bilinear pooling [4], which is

$$\begin{aligned} \boldsymbol{\mu} &= \mathcal{T}_2(\mathcal{T}_1(\mathbf{H}^T\mathbf{A})) \\ \boldsymbol{\sigma^2} &= \mathcal{T}_2(\mathcal{T}_1((\mathbf{H}\odot\mathbf{H})^T\mathbf{A}) - (\mathcal{T}_1(\mathbf{H}^T\mathbf{A}))\odot\mathcal{T}_1(\mathbf{H}^T\mathbf{A})) \end{aligned} \tag{6}$$

where $\mathcal{T}_1(x)$ is the operation of reshaping x into a vector, and $\mathcal{T}_2(x)$ includes a signed square-root step and a L2-normalization step. $\odot$ represents the Hadamard product. The output of ABP is the concatenation of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$.

It is worth noting that the proposed ABP method derives the attention map using the softmax along temporal axis to obtain the attention for each frame. And the attentive 2^{nd}-order statistics information is further exploited for aggregation, similar as statistics pooling in [21]. This is different from the existing pooling methods, such as NetVLAD [11] and learnable dictionary encoding (LDE) [10], which mainly focus on deriving Baum-Welch statistics over the channel dimension.

Compared to attentive statistic pooling [8], ABP with multiple attention heads can better capture the speaker information in a input utterance. Furthermore, ABP normalizes the length of statistics before concatenation, which is able to extract more robust embeddings.

Table 1: *Detailed configuration of the front-end extraction network.*

Layer	**Structure**	**Stride**	**Output size**
Conv1	$7\times 7,\ 16$	1×1	$L\times 35\times 16$
Res1	$\begin{bmatrix}3\times 3,\ 16\\3\times 3,\ 16\end{bmatrix}\times 2$	1×1	$L\times 35\times 16$
Trans1	$3\times 3,\ 32$	1×2	$L\times 17\times 32$
Res2	$\begin{bmatrix}3\times 3,\ 32\\3\times 3,\ 32\end{bmatrix}\times 2$	1×1	$L\times 17\times 32$
Trans2	$3\times 3,\ 64$	1×2	$L\times 8\times 64$
Res3	$\begin{bmatrix}3\times 3,\ 64\\3\times 3,\ 64\end{bmatrix}\times 2$	1×1	$L\times 8\times 64$
Trans3	$3\times 3,\ 128$	1×2	$L\times 3\times 128$
Res4	$\begin{bmatrix}3\times 3,\ 128\\3\times 3,\ 128\end{bmatrix}\times 2$	1×1	$L\times 3\times 128$
Trans4	$3\times 3,\ 128$	1×2	$L\times 1\times 128$
ABP	-	-	$1\times(128\times 2K)$
FC	$(128\times 2K)\times 128$	-	1×128
L2norm	-	-	1×128

4. Experimental setup and results

4.1. Dataset and input features

To investigate performance of the proposed system, we conducted extensive experiments using VoxCeleb1 benchmark [18] which contains over 140,000 utterances from 1251 speakers. The training set is the development portion without data augmentation, containing 1,211 speakers and the evaluation set consists of 37,720 trial pairs from 40 speakers.

The feature extraction process uses Kaldi [22]. In our implementation, 41-dimensional filter bank outputs (FBank) are used as acoustic features, obtained from 25ms windows with 10ms shift between frames. We apply mean-normalization over a sliding window of 3s, and use voice activity detection (VAD) to remove silent segments. The features from the training dataset are randomly truncated into short slices ranging from 2 to 4s. For evaluation, utterances are divided equally into 10 slices with 4s duration.

4.2. Model configuration

The detailed configuration of the front-end extraction network is summarized in Table 1, where L denotes variable-length data frames. The input layer consists of a single convolutional layer with kernel size of 7x7, stride of 1x1 and channel dimension of 16. Four residual stages include [2,2,2,2] basic blocks with 16, 32, 64, 128 channels respectively, and each basic block having 2 convolutional layers with filter sizes of 3x3 and a stride of 1x1. The transition layer comprises a convolutional layer with kernel size of 1x1 and stride of 1x2. After the four stages, the frequency dimension of the feature map is reduced to 1. For ABP, the output dimension 128 × $2K$ is varied with different attention heads K.

The identification branch is implemented by a FC layer with units equal to the number of speaker categories. We should note that when computing the AM-Softmax loss, the weight of this layer need to be normalized. The verification branch comprises two FC layers followed by sigmoid activation.

The mini-batch size for training is set to 128, containing 64 speakers with 2 utterances from each. All neural networks are implemented using the PyTorch framework [25]. The network is optimized using stochastic gradient descent (SGD) [26] with

Table 3: *Results for verification on VoxCeleb1 dataset. (AP refers to average pooling and SP refers to statistics pooling)*

System	Aggregation	Loss	Training set	Similarity	EER, %
i-vector+PLDA [18]	-	-	Voxceleb1	PLDA	8.80
x-vector [23]	SP	Softmax	Voxceleb1	cosine	11.3
				PLDA	7.1
ResNet-34 [5]	SP	Softmax	Voxceleb1	cosine	5.01
				PLDA	4.74
ResNet-34 [10]	LDE	A-Softmax	Voxceleb1	cosine	4.56
ResNet-20 [15]	AP	AM-Softmax	Voxceleb1	cosine	4.30
ResNet-50 [24]	AP	Softmax+Contrastive	Voxceleb2	cosine	4.19
Thin ResNet-34 [11]	NetVLAD	AM-Softmax	Voxceleb2	cosine	3.32
ResNet-18 (Ours)	AP	Softmax	Voxceleb1	cosine	4.58
	SP				4.19
	ABP				3.76
ResNet-18 (Ours)	ABP	AM-Softmax	Voxceleb1	cosine	3.51
Multi-task ResNet-18 (Ours)	ABP	-	Voxceleb1	verification output	**2.94**

momentum of 0.95 and weight decay of 5e-4. Each network is trained for 60 epochs with initial learning rate of 0.1, gradually decreasing to 0.0001. The durations of ramp-up and ramp-down periods are set to [0, 25] and [25, 40] epochs respectively. The performance is evaluated in terms of equal error rate (EER).

4.3. Results

4.3.1. Evaluation on different number of attention heads K

In Table 2, we study the effect of different number of attention heads K in proposed ABP method. Same as most existing deep embedding learning based SV methods, these results are obtained by using the modified ResNet-18 with Softmax loss to learn speaker embeddings first, and evaluating the verification scores with cosine distance measure. From Table 2, we can see that the EER reduces from 4.07% to 3.76% when K increases from 2 to 16. This indicates that increasing the number of attention heads can improve the effectiveness of the proposed ABP method. However, large value of K may lead to large model size and high computational complexity. In the following experiments, we only report the results with $K = 16$, considering the trade-off between effectiveness and efficiency.

4.3.2. Main results

The main results are reported in Table 3. We compared the performance of three systems including: 1) ResNet-18 with Softmax loss, 2) ResNet-18 with AM-Softmax loss, and 3) Multi-task ResNet-18. The first two systems are implemented following the existing deep embedding learning based methods, which compute the verification scores via cosine distance measure. The multi-task ResNet-18 is implemented using the proposed speaker recognition method based on joint identification and verification supervisions, and the performance is evaluated according to the output of the verification branch directly.

From Table 3, we see that the proposed system outperforms all other SV methods by a large margin. The performance of the proposed ABP method is evaluated first. We can see that our ResNet-18 with ABP achieves an EER of 3.76%, which is better than 4.58% when using average pooling and 4.19% when using statistic pooling. This indicates the superiority of the ABP method. This result is also better than the systems in [5, 10, 15, 23], demonstrating the effectiveness for embedding learning of our modified ResNet-18 architecture and pooling method.

Thanks to the role of the margin parameter, ResNet-18 with AM-Softmax loss achieves an EER of 3.51%, which is a slight improvement compared with the Softmax model. The EER is further reduced to 2.94% with Multi-task ResNet-18, outperforming almost all other deep embedding learning based SV systems in the same situation.

Table 2: *Results on different numbers of attention heads K.*

K	2	4	8	16
EER, %	4.07	3.91	3.82	**3.76**

5. Conclusion

In this paper, inspired by a multi-task framework, an effective speaker recognition method based on joint identification and verification supervision is proposed. Specifically, a deep architecture with convolutional feature extractor, attentive pooling and two branches of classifiers is presented. The first, an identification branch, is trained with AM-Softmax loss for speaker identity classification. The second, a verification branch, trains a discriminator with BCE loss to optimize the MI between positive and negative samples extracted from the embedding space. To balance these two losses at different training stages, a novel ramp-up/ramp-down weighting scheme is employed and, furthermore, a novel attentive bilinear pooling method is proposed. This further improves the effectiveness of embeddings. Experiments conducted on the Voxceleb1 benchmark yield exceptional results, demonstrating the effectiveness of the proposed model for the SV task.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. U1613211), the Leading Plan of CAS (XDC08030200), and by Key Science and Technology Project of Anhui Province (Grant No. 17030901005 and No. 18030901016).

7. References

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

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

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey *et al.*, “X-vectors: Robust DNN embeddings for speaker recognition,” in *Proc. ICASSP*, 2018, pp. 5329–5333.

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

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

[6] Y. Jiang, Y. Song, I. McLoughlin, Z. Gao, and L. Dai, “An effective deep embedding learning architecture for speaker verification,” in *Proc. Interspeech*, 2019, pp. 4040–4044.

[7] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *Proc. ICASSP*, 2018, pp. 4879–4883.

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

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

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

[11] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, “Utterance-level aggregation for speaker recognition in the wild,” *Proc. ICASSP*, 2019.

[12] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, and X. Liu, “Deep speaker: an end-to-end neural speaker embedding system,” in *arXiv preprint arXiv:1705.02304*, 2017.

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

[14] Y. Jiang, Y. Song, J. Yan, L. Dai, and I. McLoughlin, “Triplet-center loss based deep embedding learning method for speaker verification,” in *Proc. APSIPA*, 2019.

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

[16] Y. Liu, L. He, and J. Liu, “Large margin softmax loss for speaker verification,” in *Proc. Interspeech*, 2019, pp. 2873–2877.

[17] M. Ravanelli and Y. Bengio, “Learning speaker representations with mutual information,” in *Proc. Interspeech*, 2019, pp. 1153–1157.

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

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. ICASSP*, 2018, pp. 5334–5338.

[20] P. Brakel and Y. Bengio, “Learning independent features with adversarial nets for non-linear ica,” in *arXiv preprint arXiv:1710.05050*, 2017.

[21] 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.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget *et al.*, “The Kaldi speech recognition toolkit,” in *Proc. ASRU*. IEEE Signal Processing Society, 2011.

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

[24] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” in *Proc. Interspeech*, 2018, pp. 1086–1090.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan *et al.*, “Automatic differentiation in pytorch,” 2017.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, 1998, pp. 2278–2324.