



End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification

Hee-Soo Heo, Jee-weon Jung, IL-Ho Yang, Sung-Hyun Yoon, Hye-jin Shim, and Ha-Jin Yu[†]

School of Computer Science, University of Seoul, South Korea

zhasgone@naver.com, jeewon.leo.jung@gmail.com, heisco@hanmail.net,
ysh901108@naver.com, shimhz6.6@gmail.com, hjyu@uos.ac.kr

Abstract

In recent years, speaker verification has primarily performed using deep neural networks that are trained to output embeddings from input features such as spectrograms or Mel-filterbank energies. Studies that design various loss functions, including metric learning have been widely explored. In this study, we propose two end-to-end loss functions for speaker verification using the concept of speaker bases, which are trainable parameters. One loss function is designed to further increase the inter-speaker variation, and the other is designed to conduct the identical concept with hard negative mining. Each speaker basis is designed to represent the corresponding speaker in the process of training deep neural networks. In contrast to the conventional loss functions that can consider only a limited number of speakers included in a mini-batch, the proposed loss functions can consider all the speakers in the training set regardless of the mini-batch composition. In particular, the proposed loss functions enable hard negative mining and calculations of between-speaker variations with consideration of all speakers. Through experiments on VoxCeleb1 and VoxCeleb2 datasets, we confirmed that the proposed loss functions could supplement conventional softmax and center loss functions.

Index Terms: Speaker verification, end-to-end loss, metric learning, speaker embedding

1. Introduction

In recent years, several studies have reported superior results using deep neural networks (DNNs) for extracting speaker embeddings compared to conventional state-of-the-art i-vector-based [1] speaker verification systems [2–8]. Therefore, several recent studies have mainly focused on designing loss functions to train DNNs to make them suitable for speaker verification. Wan *et al.* proposed a generalized end-to-end (GE2E) loss function based on centroids, which are the average embeddings for each speaker, to train DNNs with higher generalization performance [5]. Li *et al.* applied a loss function based on angular softmax, which was proposed for face recognition [9], to create an angular margin between speakers in an embedding space [4].

The conventional studies on loss functions mentioned above do not address the following two problems. The first problem is that conventional loss functions only consider a limited number of speakers according to mini-batch composition. In the process of repeatedly training DNNs with mini-batches of a small size, the parameters of a network could be biased to

only the speakers included in one mini-batch. The second problem is that excessive overhead occurs in performing hard negative mining, which is important in metric-learning-based loss functions [10]. Hard negative mining is known to have a significant impact on the performance of metric learning. However, it is usually performed at regular intervals because of practical issues. Ideally, hard negative mining should be conducted for each mini-batch. This is because hard negative samples will change as weight parameters are updated every mini-batch. Although GE2E has partially solved these problems, there is a limitation that only few speakers can be considered by hard negative mining in GE2E.

In this paper, we propose loss functions based on speaker bases to handle these problems. The speaker bases refer to the column vectors of the weight matrix of the output layer. This definition stems from the fact that since the column vectors represent speakers in embedding space (output of the last hidden layer), each column vector can be considered as a basis for that speaker. This concept can be considered rather a general approach because it can be applied to any DNN-based speaker embedding extraction system that comprises a fully-connected code layer. We expect that it would be possible to train all speakers simultaneously and perform hard negative mining in every mini-batch using the loss function based on the speaker bases.

2. Related works

In this section, we introduce various existing loss functions that can be used to train speaker verification systems. These loss functions are already successfully applied to speaker verification and the face recognition field.

2.1. Softmax-based loss function

The softmax-based loss function is widely used to train DNNs for identification purposes. Generally, when the softmax-based loss function is exploited for speaker verification, the output of the last hidden layer is used as the embedding of each utterance after training DNNs. The softmax loss function is calculated as:

$$\mathcal{L}_S = - \sum_{i=1}^M \log \frac{\exp(W_{y_i}^T \mathbf{e}_i + b_{y_i})}{\sum_{j=1}^N \exp(W_j^T \mathbf{e}_i + b_j)}, \quad (1)$$

where \mathbf{e}_i and y_i denote the embedding (output of the last hidden layer) of the i 'th utterance and the corresponding speaker label, respectively, M is the number of utterances, N is the number of speakers in the training set, $W = [W_1, W_2, \dots, W_N]$ and $\mathbf{b} = [b_1, b_2, \dots, b_N]$ are the weight matrix and the bias vector of the output layer, respectively, and $\exp(\cdot)$ is the exponential function.

[†] Corresponding author

This work was supported by the Technology Innovation Program (10076583, Development of free-running speech recognition technologies for embedded robot system) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)

2.2. Center loss function

The center loss function was proposed to reduce within-class variations to supplement the softmax-based loss function [11]. To reduce within-class variations, loss is calculated based on the mean squared error between the embeddings of each utterance and the center embedding of the corresponding speaker. This loss function was successfully applied in the field of face recognition, and high performance improvement was reported. The center loss function, defined in equation (2), is used in conjunction with the conventional softmax-based loss function in most cases.

$$\mathcal{L}_C = \frac{\lambda}{2} \sum_{i=1}^M \|e_i - c_{y_i}\|_2^2, \quad (2)$$

where c_i is the center embedding of the i' th speaker and λ is the weight factor of the center loss function. The center embedding of each speaker in the center loss function is not trained based on gradient descent, like other parameters of DNNs. Rather, it is trained by moving the center embedding by a scalar α based on the delta center value calculated using the following formula:

$$\Delta c_k = \frac{\sum_{i=1}^M \delta(y_i = k) \cdot (c_k - e_i)}{1 + \sum_{i=1}^M \delta(y_i = k)}, \quad (3)$$

where $\delta(\text{condition}) = 1$ if the *condition* is satisfied; otherwise, $\delta(\text{condition}) = 0$.

2.3. Additive margin loss function

The additive margin softmax (AMsoftmax) loss function was proposed to replace the inner product operation of the softmax-based loss function with the cosine similarity operation [9] and widen the margin between each class in an embedding space [12]. The AMsoftmax loss function is calculated based on the cosine similarity, $\cos(\cdot, \cdot)$, so that the embedding between each speaker has an additional margin of m , as follows:

$$\mathcal{L}_{AMS} = - \sum_{i=1}^M \log \frac{\exp(s \cdot (\cos(W_{y_i}, e_i) - m))}{\exp(s \cdot (\cos(W_{y_i}, e_i) - m)) + R_i}, \quad (4)$$

$$R_i = \sum_{j=1, j \neq y_i}^N \exp(s \cdot \cos(W_{y_j}, e_i)), \quad (5)$$

where s is a scaling factor for stabilizing the training process of cosine similarity-based loss. This loss function is an improved version of the angular softmax function [9] and has been successfully applied to speaker recognition systems [4, 13, 14].

2.4. Generalized end-to-end loss function

The GE2E loss function, a recent one of the advanced versions of triplet loss, was proposed to reduce the distance between the embeddings of each utterance and the centroid embeddings of the corresponding speaker while increasing the distance from the centroid embeddings of other speakers [15]. The most significant characteristic of the GE2E loss function is that it does not calculate the distance between samples but calculates the distance between centroids by averaging the embeddings from the same speaker. Wan, Li *et al.* assumed that higher generalization performance could be achieved through a distance comparison with centroid embeddings [15]. For this purpose, the distance between the embedding of the i' th utterance of the j' th speaker and the centroid of the k' th speaker, $S_{j_i,k}$, is calculated as follows:

$$S_{j_i,k} = w_{score} \cdot \cos(e_{j_i}, \hat{c}_k) + b_{score}, \quad (6)$$

where w_{score} and b_{score} are the trainable parameters for scaling and shifting scores. It is important to note that the centroid \hat{c}_k , in equation (6), is different from the center embedding, c_k , in equation (2). Centroid embedding is calculated utilizing the embeddings of each speaker as follows:

$$\hat{c}_k = \frac{1}{M_k} \sum_{i=1}^{M_k} e_{ki}, \quad (7)$$

where M_k is the number of utterances of the k' th speaker. The GE2E loss function is calculated based on $S_{j_i,k}$, as follows:

$$\mathcal{L}_G = \sum_{j,i} 1 - \sigma(S_{j_i,j}) + \max_{\substack{1 < k < N \\ k \neq j}} \sigma(S_{j_i,k}), \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function for stabilizing training. In the GE2E loss function, hard negative mining is performed by selecting the largest value among the scores of negative pairs. It is important to note that it is required to construct one mini-batch with a few utterances of speakers for calculating the loss defined by equation (6). This is because the centroid, \hat{c}_k , in equation (6) is calculated from multiple utterances of each speaker. This requirement limits the mini-batch configuration, thereby considerably reducing the number of speakers in a mini-batch. For example, if one configures a mini-batch of size 100 and each mini-batch includes five utterances per speaker, only 20 speakers are included in a mini-batch. This may be too small, considering the number of whole speakers in the dataset for speaker recognition [16, 17].

3. Proposed loss functions based on class bases

The core idea behind the proposed loss functions is that the weight matrix between the last hidden layer and output layer in softmax-based loss function can be interpreted as a set of bases where each basis vector represents a class, i.e. a speaker. For example, in speaker identification task, to calculate the softmax loss function for 1000 speakers from 128-dimensional embedding, a weight matrix of size [128, 1000] is required. This weight matrix can be interpreted as a set of 128-dimensional vectors, which represent each speaker. With this interpretation, these basis vectors were trained to replace each speaker's centroid. Therefore, it is possible to train the system using the entire classes at every mini-batch regardless of its size.

We propose an additional loss function to maximize the between-class variation based on the proposed class basis interpretation as the following equation

$$\mathcal{L}_{BC} = \sum_{i=1}^N \sum_{j=1, j \neq i}^N \cos(W_i, W_j), \quad (9)$$

where N is the number of classes, and W_i is the basis of the i' th class. The conventional loss based on categorical cross entropy also can increase the between class variation. However, the conventional loss can consider only the limited number of classes depending on the size of the mini-batch. On the other hand, \mathcal{L}_{BC} can consider the entire classes simultaneously. We can also interpret the behavior of \mathcal{L}_{BC} as to distribute the bases equally in the embedding space. We expect that the proposed loss function \mathcal{L}_{BC} is complementary to the conventional center

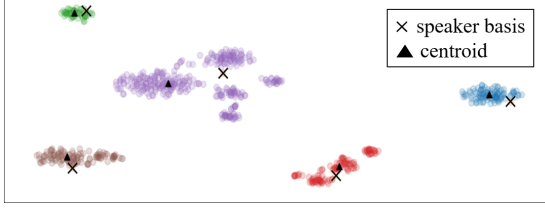


Figure 1: Embeddings visualized using the *t*-SNE algorithm [19]. Five different colors refer to five randomly selected speakers from the training set. Triangles are the average embeddings of each speaker. *x*'s are the speaker bases.

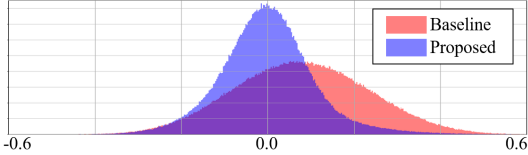


Figure 2: Histogram of the impostor scores calculated according to the proposed loss function, \mathcal{L}_{BC} (*x*-axis: $\cos(\hat{c}_i, \hat{c}_j)|_{i \neq j}$, *y*-axis: density).

loss function, which only considers within-class variations (see Table 2).

Hard negative mining, the method to select samples or pairs that are likely to be mistaken in the training set, is known as one of the important factors that determine the performance of metric learning [18]. This is because the hard negative samples have a dominant influence on the decision boundaries of the classes. In conventional metric learning, hard negative mining is typically performed in a separate phase. However, it is difficult to increase the frequency of hard negative mining because of the overhead in the phase. The GE2E loss function deals with the same problem in a manner to the proposed loss function, but the number of speakers included in negative mining is quite limited. To address this difficulty, we propose another objective function based on the class basis interpretation for performing hard negative mining for all speakers. The proposed loss function is defined as:

$$\mathcal{L}_H = \sum_{i=1}^M \sum_{W_h \in \mathcal{H}_i} \log(1 + \exp(\cos(W_h, e_i) - \cos(W_{y_i}, e_i))), \quad (10)$$

where e_i and W_{y_i} denote the i 'th utterance and the basis of the corresponding speaker, respectively, and \mathcal{H}_i is the set of the top H speaker bases with large $\cos(W_h, e_i)|_{h \neq y_i}$ values.

The main purpose of the loss \mathcal{L}_H is to reduce the negative similarities, represented as $\cos(W_h, e_i)$, while increasing the positive similarities, represented as $\cos(W_{y_i}, e_i)$. Exponential function $\exp(\cdot)$ is applied with the perspective to increase the gradient of the samples with large loss and decrease the gradient of the samples with small loss, simultaneously. The additional $1 +$ term in the $\log(x)$ function limits the value of x to greater than one. This is because the $\log(x)$ function has too small a value and an overly large gradient when the value of x is close to zero. The proposed negative mining objective function enables negative mining on every mini-batch, considering entire classes.

Figure 1 illustrates the embeddings, centroids, and speaker bases extracted from the trained DNN using the proposed loss functions ($\mathcal{L}_{BC} + \mathcal{L}_H$), from the utterances of five randomly

Table 1: DNN architecture (l : length of input sequence).

layer	output shape	kernel size	stride
Conv1	$l \times 64 \times 16$	7×7	1×1
Res1	$l \times 64 \times 16$	3×3	1×1
Res2	$(l/2) \times 32 \times 32$	3×3	2×2
Res3	$(l/4) \times 16 \times 64$	3×3	2×2
Res4	$(l/8) \times 8 \times 128$	3×3	2×2
Pool	128	Global	Global

selected speakers. The centroids were calculated by averaging all embeddings of each speaker after the training process. Note that these centroids are ideal and shown for comparison purpose only and calculating centroids at every mini-batch is impractical. This is because all utterances of a speaker should be considered for calculating one centroid. The figure shows that each speaker can be represented by a speaker basis.

Figure 2 shows the histogram of impostor scores, which are calculated as $\cos(\hat{c}_i, \hat{c}_j)|_{i \neq j}$, to confirm the effect of the proposed loss function, \mathcal{L}_{BC} . We compared the difference between the baseline with (trained by $\mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_{BC}$) and without (trained by $\mathcal{L}_S + \mathcal{L}_C$) applying the proposed loss function. The result demonstrates that the impostor scores of the training set are reduced by the proposed loss function and between-speaker variations are increased compared to the center loss function (baseline).

4. Experiments

We used VoxCeleb1 dataset which comprises 1,251 speakers with approximately 330 hours of utterances, following the guideline provided in [16]. The guideline assigns 1,211 speakers as the training set and 40 speakers as the evaluation set. We implemented the DNNs based on Keras with TensorFlow as the back-end [20–22]. We used Kaldi for acoustic feature extraction [23].

4.1. Experimental configuration

64-dimensional Mel-filterbank energy features were extracted using a 25-ms hamming window with a 10-ms shift [24]. Mean normalization was applied over a 3-s sliding window. ResNet-34 [25, 26] was modified as shown in Table 1 and used for extracting 128-dimensional speaker embeddings. We used leaky rectified linear unit [27] as the activation function. The Adam optimizer with a learning rate of 0.001 was utilized with a mini-batch size of 100. In our experiments, the performance with the loss function defined by the inner product was degraded by weight decay whereas the performance with the loss function defined by the cosine similarity was improved by weight decay. Table 2 demonstrates the hyper-parameters and the EER of baseline and proposed systems.

4.2. Results and analysis

Results of the implementation in this study showed that the DNN trained using the center loss showed the lowest EER among other losses discussed in Section 2. The GE2E loss function, which is expected to have high performance, exhibited a relatively high EER. This result is interpreted as a phenomenon caused by a fixed mini-batch size owing to practical issues such as GPU memory. In particular, if the mini-batch size was fixed at 100 and five utterances for each speaker were included, one mini-batch would contain only 20 speakers. This

Table 2: Performances in terms of EER and hyper-parameters of various systems (\mathcal{L}_S : softmax-based loss, \mathcal{L}_C : center loss, \mathcal{L}_{AMS} : additive margin softmax, \mathcal{L}_G : generalized end-to-end loss, \mathcal{L}_{BC} : proposed between-speaker variations loss, \mathcal{L}_H : proposed hard negative mining loss).

System	Loss	hyper-parameters	EER (%)
i-vector PLDA reported in [16]	-	-	8.8
Metric learning reported in [16]	-	-	7.8
Softmax loss (our implementation)	\mathcal{L}_S	-	7.78
Center loss (our implementation)	$\mathcal{L}_S + \mathcal{L}_C$	$\lambda = 0.001, \alpha = 0.5$	6.55
AM softmax (our implementation)	\mathcal{L}_{AMS}	$s = 5, m = 0.35, \text{weight decay}(0.0001)$	7.31
GE2E (our implementation)	\mathcal{L}_G	5 utterances for each speaker, weight decay(0.0001)	10.65
Proposed 1	$\mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_{BC}$	$\lambda = 0.001, \alpha = 0.5$	5.96
Proposed 2	$\mathcal{L}_H + \mathcal{L}_{BC}$	$H = 100, \text{weight decay}(0.0001)$	5.55

is an extremely limited number considering the number of all speakers. Based on the center loss function, which showed the highest performance among the conventional loss functions, we applied the proposed loss function and compared the performances. First, the loss function, \mathcal{L}_{BC} , defined using equation (9) with the center loss function could reduce the EER by approximately 9% relatively. This result indirectly shows that between-speaker variations were increased by the proposed loss function, \mathcal{L}_{BC} . In addition, the error was reduced by 6% by replacing the center loss function by the proposed loss function, \mathcal{L}_H . Finally, the proposed loss function reduced the error by 15% relatively. Based on these results, we found that it is possible to design an effective loss function for speaker verification with the proposed speaker bases.

4.3. Additional validation using VoxCeleb2

We conducted experiments using the VoxCeleb2 dataset to further verify the proposed loss functions. In this configuration, VoxCeleb2, with 6,112 speakers, is used as training set, and the entire VoxCeleb1, with 1,251 speakers, is used as test set. The DNNs trained by the proposed loss functions were evaluated using three different trials lists introduced in [28]: (1) the original trials list of VoxCeleb1, (2) VoxCeleb1-E, and (3) VoxCeleb1-H. As the amount of training increased, we modified the architecture of DNNs as shown in Table 3. The b-vector system was implemented for back-end scoring instead of cosine similarity [29]. In this system, a b-vector is extracted from two embeddings using element-wise binary operations such as addition, subtraction, and multiplication. A 3072-dimensional (1024×3) b-vector is extracted from two 1024-dimensional embeddings. The b-vector system was first introduced for scoring the i-vectors. Then, it has been shown that the b-vector based system could be efficient classifier of embeddings from DNNs in speaker verification task [6, 30]. In our experiments, the b-vector based system comprises three fully-connected lay-

Table 3: DNN architecture in the experiments on VoxCeleb2 (l : length of input sequence).

layer	output shape	kernel size	stride
Conv1	$l \times 32 \times 64$	7×7	1×2
Res1	$l \times 32 \times 64$	3×3	1×1
Res2	$(l/2) \times 16 \times 128$	3×3	2×2
Res3	$(l/4) \times 8 \times 256$	3×3	2×2
Res4	$(l/8) \times 4 \times 512$	3×3	2×2
Pool	512	Global	Global
Dense	1024	512×1024	-

Table 4: Performance comparison of the experiments using VoxCeleb2 (\mathcal{L}_S : softmax-based loss, \mathcal{L}_C : center loss, \mathcal{L}_{BC} : proposed between-speaker variations loss, \mathcal{L}_H : proposed hard negative mining loss, orgVox1: original test lists of VoxCeleb1, Vox1-E: extended test lists based on all speakers of VoxCeleb1, Vox1-H: extended and hard test lists based on all speakers of VoxCeleb1).

Loss	orgVox1	Vox1-E	Vox1-H
$\mathcal{L}_S + \mathcal{L}_C$	2.99	3.01	5.19
$\mathcal{L}_{BC} + \mathcal{L}_H(H = 100)$	2.75	2.71	4.53
$\mathcal{L}_{BC} + \mathcal{L}_H(H = 50)$	2.66	2.43	4.08

ers, each with 512 nodes, and an output layer with two nodes. The two nodes of the output layer indicate whether the two embedding inputs are from the same speaker or not, respectively.

Table 4 shows the results of an additional validation. For the validation, we compared the performances of the system trained by \mathcal{L}_S (the best performance among the conventional losses) and the proposed $\mathcal{L}_{BC} + \mathcal{L}_H$. Experimental results demonstrate that the proposed loss can replace the conventional loss while reducing the error by 21 % relatively.

5. Conclusions

In this study, we proposed an interpretation of the weight matrix of the output layer as class bases. We applied the proposed end-to-end loss functions using this interpretation for speaker verification. The proposed loss function comprises \mathcal{L}_{BC} for increasing between-speaker variations and \mathcal{L}_H for hard negative mining. The biggest advantage of the proposed loss functions is that regardless of the composition of the mini-batch, all speakers can be considered simultaneously. The experimental results obtained for the VoxCeleb showed that the error in the proposed loss function was reduced by approximately 15% relatively compared with the error in the conventional loss functions. In addition, we found that the proposed loss function could replace the conventional loss functions.

In this paper, we intensively analyzed the performances depending on the various loss functions without modification of the DNN architecture. As a future work, we plan to apply the proposed loss functions to various DNN architectures including the x-vector system [8]. In addition, we also have a plan to apply the proposed interpretation regarding the class basis of the output layer into other domain tasks such as face recognition.

6. References

- [1] 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, 2011.
- [2] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.
- [3] 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.
- [4] L. Yutian, G. Feng, O. Zhijian, and S. Jiasong, "Angular softmax loss for end-to-end speaker verification," *Proceedings of International Symposium on Chinese Spoken Language Processing*, 2018.
- [5] L. Wan, Q. Wang, A. Papir, and I. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.
- [6] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5349–5353.
- [7] —, "Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3583–3587.
- [8] 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.
- [9] L. Weiyang, W. Yandong, Y. Zhiding, L. Ming, R. Bhiksha, and S. Le, "Sphereface: Deep hypersphere embedding for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 1.
- [10] S. Florian, K. Dmitry, and P. James, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [11] W. Yandong, Z. Kaipeng, L. Zhifeng, and Q. Yu, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.
- [12] W. Feng, C. Jian, L. Weiyang, and L. Haijun, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [13] J. A. C. Nunes, D. Macêdo, and C. Zanchettin, "Additive margin sincnet for speaker recognition," *arXiv preprint arXiv:1901.10826*, 2019.
- [14] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6046–6050.
- [15] W. Li, W. Quan, P. Alan, and M. I. Lopez, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.
- [16] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.
- [17] M. F. Alvin and G. S. Craig, "The nist 2010 speaker recognition evaluation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.
- [18] B. Harwood, B. Kumar, G. Carneiro, I. Reid, T. Drummond *et al.*, "Smart mining for deep metric learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2821–2829.
- [19] L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, 2008.
- [20] F. Chollet *et al.*, "Keras," <https://github.com/keras-team/keras>, 2015.
- [21] A. Martín, A. Ashish, B. Paul, B. Eugene *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: <http://download.tensorflow.org/paper/whitepaper2015.pdf>
- [22] A. Martin, B. Paul, C. Jianmin, C. Zhifeng, D. Andy, D. Jeffrey, D. Matthieu, G. Sanjay, I. Geoffrey, I. Michael, K. Manjunath, L. Josh, M. Rajat, M. Sherry, M. G. Derek, S. Benoit, T. Paul, V. Vijay, W. Pete, W. Martin, Y. Yuan, and Z. Xiaoqiang, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>
- [23] P. Daniel, G. Arnab, B. Gilles, B. Lukas, G. Ondrej, G. Nagendra, H. Mirko, M. Petr, Q. Yanmin, S. Petr *et al.*, "The kaldı speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.
- [24] 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.
- [25] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
- [27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [28] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.
- [29] H. S. Lee, Y. Tso, Y. F. Chang, H. M. Wang, and S. K. Jeng, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1660–1664.
- [30] H.-S. Heo, I.-H. Yang, M.-J. Kim, S.-H. Yoon, and H.-J. Yu, "Advanced b-vector system based deep neural network as classifier for speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5465–5469.