AIMS Mathematics

AIMS Mathematics, 7(4): 6381–6395.
DOI: 10.3934/math.2022355
Received: 29 September 2021
Revised: 29 December 2021
Accepted: 06 January 2022
Published: 19 January 2022

http://www.aimspress.com/journal/Math

Research article

DeltaVLAD: An efficient optimization algorithm to discriminate speaker embedding for text-independent speaker verification

Xin Guo[1], Chengfang Luo[2], Aiwen Deng[2] and Feiqi Deng[2,*]

[1] Guangdong Communication Polytechnic, Guangzhou 510650, China
[2] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China

* **Correspondence:** Email: aufqdeng@scut.edu.cn; Tel: +13922202550.

Abstract: Text-independent speaker verification aims to determine whether two given utterances in open-set task originate from the same speaker or not. In this paper, some ways are explored to enhance the discrimination of embeddings in speaker verification. Firstly, difference is used in the coding layer to process speaker features to form the DeltaVLAD layer. The frame-level speaker representation is extracted by the deep neural network with differential operations to calculate the dynamic changes between frames, which is more conducive to capturing insignificant changes in the voiceprint. Meanwhile, NeXtVLAD is adopted to split the frame-level features into multiple word spaces before aggregating, and subsequently perform VLAD operations in each subspace, which can significantly reduce the number of parameters and improve performance. Secondly, the margin-based softmax loss function and the few-shot learning-based loss function are proposed to be combined for more discriminative speaker embeddings. Finally, for a fair comparison, the experimental results are performed on Voxceleb-1 showing superior performance of speaker verification system and can obtain new state-of-the-art results.

Keywords: speaker verification; text-independent; difference; DeltaVLAD; margin-based softmax loss function; few-shot learning-based loss function
Mathematics Subject Classification: 00A67, 33B99

1. Introduction

Speaker recognition (SR) is a key technology for intelligent interaction. In view of free accessibility and improved arithmetic capability of large databases such as the VoxCeleb [1,2], SR has made good progress in recent years. Voiceprint is a behavioral feature with physiological characteristics, unlike speech recognition, speaker recognition takes no account of the meaning of words, and it just focuses more on the speaker's identity.

Speaker recognition (SR) is mainly divided into speaker verification (SV) and speaker identification (SI) [3]. SV is a 1:1 task to confirm whether two voices originate from the same person, while SI is a 1:n task to confirm whether the target speaker's voice appears in the given samples. Meanwhile, SR can be divided into text-dependent and text-independent depending on the content [4], between which the former is more common and more challenging in the real world. We must specify that the scope of research is text-independent in this paper.

The Gaussian mixture model-universal background model (GMM-UBM) made the significant contribution to the development of SR technology from the laboratory to the application [5], but GMM-UBM was limited in the ability to handle complex data of a real scene, therefore we will discuss the model based on deep learning in this paper. At present, SV based on deep learning are of three branches: DNN/i-vector-based model [6], embedding-based model [7] and end-to-end model [8].

Due to the further development of deep learning and the strong data-fitting ability of deep learning models, the voiceprint research has gradually shifted to deep learning for feature extraction. The input of end-to-end model is a waveform with the model requiring performing a large number of calculations, making the training harder, and the embedding-based speaker recognition is more common. This paper is dedicated to research on the embedding-based speaker recognition.

In recent years, most researchers in the speaker recognition research community have focused on the feature encoding layer and have proposed many excellent feature encoding methods， such as ANF [9], ABP [10], VBA [11] and Segment Aggregation [12], which bring a great improvement in the speaker recognition models. As shown in these works, one of the contributions of this paper is to propose a new feature encoding method called Delta-VLAD. The dynamic relationship between preceding and following frames in frame-level features can also indicate the identity of the speaker, therefore, we capture such dynamic features that can indicate the identity of the speaker by calculating the delta coefficients of the context in frame-level features in Delta-VLAD. Experimental results show the EER of the method with Delta-VLAD outperforms the baseline model by 25%.

In addition to the research on feature encoding methods, the research on loss functions has received more attention in the community. In particular, the aggregation of the prototypical loss function and the margin-based softmax loss function were proposed in our previous work [13], which made a large performance improvement. In this paper, we keep exploring the application of the combined loss function in a SV model, and make further performance improvements by adjusting the corresponding weight β.

In summary, the main contributions of this paper are as follows:

(1) The Delta-VLAD is proposed. It captures the dynamic relationship between the preceding and following frames in frame-level features, which models the nature of dynamic features in acoustic features.

(2) A combined loss function is further explored. The combination of prototypical loss function based on the few-shot learning framework and margin-based softmax loss function is further explored

in SV domain, and the advantages of such combined loss functions are fully demonstrated.

(3) The experimental results indicate that the proposed methods achieve SOTA performance on the voxceleb1 dataset.

2. Related works

How to make full use of the voiceprint feature information is the key for SR. Generally speaking, there are three modules for optimizing SR system: frame-level feature extractor, the coding layer and loss function.

In the field of deep learning for SR, the main backbone networks are MP [9], DCNN [14], TDNN [7,15], and RNN [16]. They are used as feature extractors for extracting elements that characterize the speaker. To reduce model parameters and increase convergence speed, the fast ResNet34 network [17] is proposed and achieved excellent performance. Meanwhile, the attentive SE block is also used to pay more attention to the contributions of different local information.

After going through the frame-level feature extractor, the output is still frame-level features with its length related to that of the input acoustic features. The role of the encoding layer is to map the variable-length frame-level features into fixed-dimensional utterance-level embeddings for the purposes of facilitating subsequent classification and discrimination. The current methods for encoding frame level features at the pooling layer are mainly classified into statistical-based encoding methods and dictionary-based encoding methods. The statistical-based methods include AP [1], SP [7], SAP [17], ASP [18], etc. The dictionary-based encoding methods include LDE [17], GhostVLAD [19], NetVLAD [20], etc. Notably, NeXtVLAD [21] has a powerful capability in encoding feature, it can split frame-level features into multiple word spaces before aggregating and then perform VLAD operation in each subspace, significantly reducing the number of parameters as well as enhancing performance. All of them can project variable-length utterances into fixed-length speaker characterizing embeddings, but they use only the static features of speech. [22] shows more efficient results obtained by simply computing the pixel differences between two adjacent frames in the feature space and the corresponding PA, and focusing on simulating small displacement at the motion boundary.

The additional margin-based softmax function was introduced to the field of speaker recognition in [7,23], and [24,9,25] used a prototype loss function to train text-independent speaker recognition models, where the speaker optimized the metric space by finding the class prototype closest to the target recognition sample. [26,27] combined prototype loss function and softmax loss function to train a speaker recognition model, which could be used to produce more discriminative speaker feature vectors. Unlike the above cases, our loss function is a combination of metric loss function and classification loss function, in other words, a combination of a prototype loss function based on the cosine similarity and a softmax loss function based on additional margin, and both of which collaborate and complement each other and thus further augment the discrimination of speakers.

3. Methods

In this section, we first systematically introduce the structural framework used in the work, and then specify the differential coding strategy and the combined loss function.

3.1. System architecture

The framework structure of the SV task is shown in Figure 1.

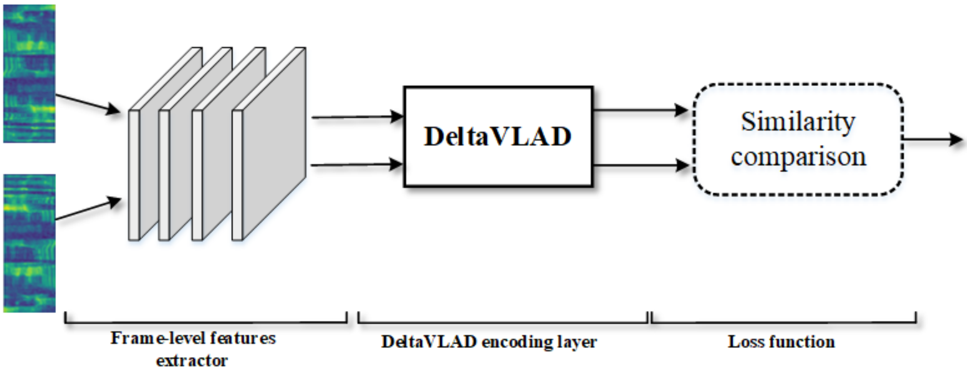

Figure 1. Framework of speaker verification model.

It consists of three modules: the frame-level feature extractor, the DeltaVLAD encoding layer, and the similarity comparison module. The two audio spectrograms are passed through the same feature extractor to form two of their own frame-level features, which are fed into the coding layer for conversion to their respective utterance-level features, and the two representations are compared for similarity to determine whether they belong to the same person's voice.

(1) Frame-level feature extractor. This network consists of a fast ResNet34 network integrated squeeze-and-excitation (SE) block [28]. The structure of the backbone network as a feature extractor is shown in Table 1.

Table 1. Structure of Frame-level features extractor.

Layer	Input Fbank(40×D)	Output size
	Conv2d,7×7 ,16	16×20×D
Block1	$\begin{bmatrix}3\times3,\text{conv},16\\3\times3,\text{conv},16\end{bmatrix}\times3$, SE layer	16×20×D
Block2	$\begin{bmatrix}3\times3,\text{conv},32\\3\times3,\text{conv},32\end{bmatrix}\times4$, SE layer	32×10×D/2
Block3	$\begin{bmatrix}3\times3,\text{conv},64\\3\times3,\text{conv},64\end{bmatrix}\times6$, SE layer	64×5×D/4
Block4	$\begin{bmatrix}3\times3,\text{conv},128\\3\times3,\text{conv},128\end{bmatrix}\times3$, SE layer	128×5×D/4
	Conv2d,1×1, 5	128×1×D/4

Arbitrary speech duration is accepted as the input, and arbitrary lengths at the frame level are generated. The Fast-ResNet34 structure is the same as the structure of the original 34-layer ResNet. However, to reduce the computational cost, only a quarter of the channels of the original model is used in each residual block of Fast-ResNet34.

(2) The coding layer. It is used to map variable-length frame-level features to fixed-length feature vectors. Inspired by the methods in literature [21,22], this paper proposes a new coding strategy: the DeltaVLAD technique. It extracts the first-order difference and second-order difference between features and neighboring features as the input to capture the dynamic relationship, which is one of the main contributions of this work.

(3) Loss function. The function in this subsection use the combined function, as shown in Figure 2, where the classification function can enlarge the inter-class distance and the metric function is designed to optimize the measurement space [29]. The combined loss function can hit the mark of optimizing the metric space and increasing the discrimination of the feature space, which is another main contribution of this work.

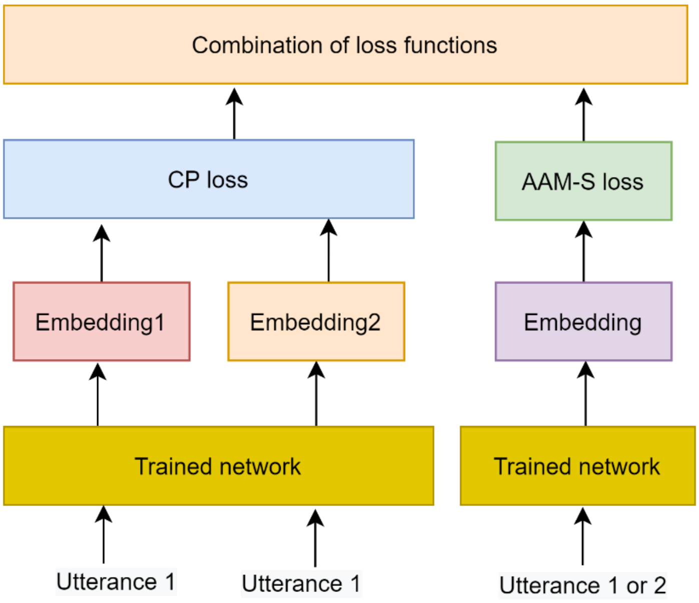

Figure 2. The combination of loss function.

3.2. DeltaVLAD encoding algorithm

The coding layer in SV is designed to encode some variable-length frame-level features and obtain a fixed-dimension utterance-level feature vector. The DeltaVLAD is a new coding method that takes full advantage of difference and NeXtVLAD.

3.2.1. Difference

Difference is a classical mathematical tool due to simple and easy to use for many applications. Difference algorithm is also an efficient global optimization algorithm, and the formula is expressed as follows:

$$\Delta^{t}=\frac{\sum_{a=1}^{A}a(f_{t+a}-f_{t-a})}{2\sum_{a=1}^{A}a^{2}}\ ,\tag{1}$$

$$a=\frac{W_L-1}{2},\tag{2}$$

where, A represents the order, $W_L$represents the window length of the voiceprint. The differential representation between frame levels in speaker recognition is shown as Figure 3.

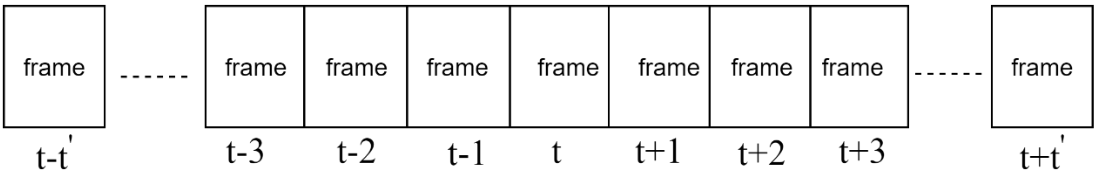

Figure 3. Schematic diagram of SV frame.

When $A=1$, the formula can be expressed as:

$$\Delta^{t}=\frac{f_{t+1}-f_{\ t-1}}{2}\ .\tag{3}$$

When $A=2$, the formula can be expressed as:

$$\begin{aligned}\Delta^{t}&=\frac{f_{t+1}-f_{\ t-1}+2(f_{t+2}-f_{t-2})}{10}\\&=\frac{1}{5}\cdot\frac{f_{t+1}-f_{\ t-1}}{2}+\frac{4}{5}\cdot\frac{f_{t+2}-f_{t-2}}{4}\ .\end{aligned}\tag{4}$$

When $A=3$, the formula can be expressed as:

$$\begin{aligned}\Delta^{t}&=\frac{f_{t+1}-f_{\ t-1}+2(f_{t+2}-f_{t-2})+3(f_{t+3}-f_{t-3})}{28}\\&=\frac{1}{14}\cdot\frac{f_{t+1}-f_{\ t-1}}{2}+\frac{4}{14}\cdot\frac{f_{t+2}-f_{t-2}}{4}+\frac{9}{14}\cdot\frac{f_{t+3}-f_{t-3}}{6}\ .\end{aligned}\tag{5}$$

From the above equation, we can see that the differential formula is a form of weighted average difference. The first-order difference is the corresponding subtraction of the subsequent frame and the previous frame of the current frame, reflecting the dynamic relationship between two adjacent frames. The second-order difference is an expression between the previous frame and the next frame based on the first order difference. Similarly, the third-order difference is an expression between the previous frame and the next frame based on the second-order difference, which is a weighted average difference formula reflecting the relationship between the six frames adjacent to the current frame, and the current order accounts for the major proportion. The others are similar, using the differential calculation as the representation of the feature, it can well reflect the changing characteristics of speaker behavioral feature.

3.2.2. DeltaVLAD coding layer

NeXtVLAD is an upgraded version of NetVLAD [20], which has significantly reduced the model parameters and improved the training efficiency with the similar performance. NeXtVLAD is originally an encoding method used for the video compression, and now introduced to the SV task.

We further use a differential strategy at the encoding layer to capture the dynamic properties of the features and encode them into low-dimensional embeddings. As shown in Figure 4, the output by the feature extractor is assumed to be f^l, then the feature input of encoding layer is spliced by first-order difference and second-order difference, thus the input of encoding layer can be expressed as follows:

$$h^{f} = f^{l} \oplus f^{l}_{\Delta^{1}} \oplus f^{l}_{\Delta^{2}}\,. \qquad (6)$$

Where Δ^1 represents first-order difference，Δ^2 represents second-order difference. In addition, $\{c\}$ represents the set of clustering centers. Then, the feature vector, the output of encoding layer, can be expressed as follows:

$$y^{f} = f_{NeXtVLAD}(h^{f})\,. \qquad (7)$$

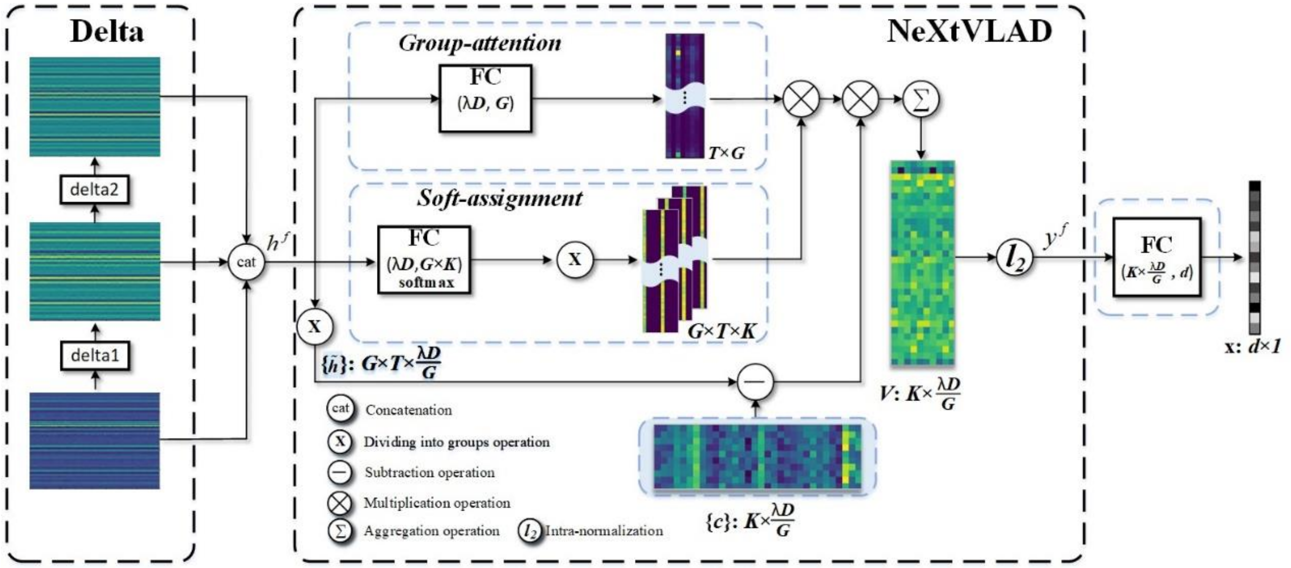

Figure 4. Structure diagram of DeltaVLAD layer.

Finally, following a fully-connected layer, the representation of the speaker feature embedding can be obtained as follows:

$$\mathrm{x} = FC(y^{f})\,. \qquad (8)$$

3.3. The proposed loss function

3.3.1. The margin-based softmax loss function

Since the softmax loss function fails to make speaker embedding intra-class compactly and inter-class separatly, the margin-based softmax loss function has been proposed in the academic community. Compared with the softmax loss function, the AM-Softmax loss function [30] and AAM-Softmax loss function [31] are learned to expand the classification boundary and increase the feature margin between different classes by introducing a cosine margin penalty to the target logit.

The AM-Softmax loss function can be expressed as follows:

$$L_{AMS}=-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos\theta_{y_i,i}-m)}}{e^{s(\cos\theta_{y_i,i}-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_{j,i})}}. \tag{9}$$

The AAM-Softmax loss function can be expressed as follows:

$$L_{AAMS}=-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{y_i,i}-m)}}{e^{s\cos(\theta_{y_i,i}-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_{j,i})}}. \tag{10}$$

However, using the loss functions alone can only forces the deep features to stay apart but fails to explicitly optimize embedding space [29]. On the other hand, training with AM-Softmax and AAM-Softmax has been proven to be challenging for they are sensitive to the value of scale and the margin in the loss function. So we need to consider other loss functions.

3.3.2. The prototypical loss function

The prototype loss function is derived from the prototype network [32]. The prototype loss function takes the similarity between instances as the input in the metric space, and its application in SV has been proven to be effective [9].

Suppose there are A speakers in the training set, and a mini-batch is randomly selected from the training set, including N different speakers, each with M utterances. The M-1 samples of each person form a set of support sets $S=\{s_{i,j}\}$, (where $1\le i\le N$， $1\le j\le M-1$), and the remaining 1 sample form a query set $Q=\{q_n\}$ (where$1\le n\le N$). $x^{s}_{i,j}$ represents the feature vector of the jth sample of the ith speaker in the support set, and x^{q}_{n} represents the feature vector of the nth speaker in the query set. The prototype of each category can be expressed as follows:

$$c_i=\frac{1}{M-1}\sum_{m=1}^{M-1}x^{s}_{i,j}. \tag{11}$$

Where c_i denotes the center of the category in the feature space of the ith speaker.

Based on the open-set metric of speaker recognition, the similarity is defined as the distance under the cosine similarity as in literature [25]:

$$S_{q_n,c_i} = \omega \cdot \cos(x_n^q, c_i) + b \,. \tag{12}$$

The similarity matrix S_{q_n,c_i} represents the similarity between the query sample q_n and the prototype center of mass c_i. Here we refer to the prototype loss function based on the cosine similarity as the cosine-prototype loss function, or CP loss function, and denote it as follows:

$$L_{CP} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{e^{S_{n,n}}}{\sum_{j=1}^{N} e^{S_{n,j}}} \,. \tag{13}$$

The model parameters can be optimized by back-propagating the gradient change of the query sample. The prototypical loss function is of strengths like simulating the few-shot learning scenario, more suitable for open-set SV tasks.

3.3.3. The combined loss function

Being an open-set task in real scenarios, SR is featured by joint functions of classification and few-shot learning. We combine the classification loss function and the prototypical loss function to learn a metric which can make similar samples close and dissimilar ones distant in the embedding space, and consequently satisfy all the requirements of open-set SV tasks.

We combine the AM-Softmax loss function (AM-S) or the AAM-Softmax loss function (AAM-S) and the cosine-based prototype loss function (CP) [13]. The AM-S and AAM-S are explicitly encouraged to make the distance between classes larger, while the CP can make it possible to find the prototype close to the target sample in the speaker feature so as to optimize the metric space. Meanwhile, the CP can also handle the situation that some samples do not appear in the training set, and greatly improve the robustness of the model. The combination of two such loss functions can train a SV network model with the best results. The optimal loss function can be trained by adjusting the hyperparameters β of the final combined loss function (take AAM-S as an example):

$$L = L_{CP} + \beta * L_{AAMS} \,. \tag{14}$$

4. Experiments

We perform experiments on VoxCeleb [1], which contains 1,251 speakers including 1,211 speakers in the training set and 40 speakers in the test set, a total of over 140,000 audios for 352 hours.

4.1. Implementation configuration

This experiment is conducted under Linux system with PyTorch framework. The experimental environment is as shown in Table 2.

The speaker features used in this experiment are 40-dimensional log Mel-Filterbank features （Fbank） with a frame length of 25 ms and a step size of 10 ms. Since the speech in the VoxCeleb-1 dataset is continuous, neither voice activity detection (VAD) is used in the experiment, and nor data augmentation operation is performed except for random sampling.

The baseline model uses fast-ResNet as the feature extractor, NetVLAD (K=10) as the feature encoding layer, and the loss function is AAM-Softmax (m=0.1), which is chosen to evaluate the performance of the present modules. The model in the experiment adopts Adam optimizer, whose weight decay rate is $5e-5$, with batchsize is set to 256 and the initial learning rate 0.005, and when the loss decrease is less than 0.01 within 5 epochs, the current learning rate is halved.

Table 2. Configuration of this experiment.

Experimental configuration	Configuration parameters
Operation system	Ubuntu 18.04.3 LTS,64 bit
CPU	Intel Xeon E5-2678 v3 2.5GHz, 48 core
GPU	Nvidia GTX 2080Ti, 11G
Memory	64G,2666MHz
Deep Learning Framework	PyTorch v1.8.0

4.2. Evaluation of combined loss functions

In this experiment, we explore the best weight β in Eq (14) under the framework with DeltaVLAD layer, and set K = 10 and G = 8. The training loss function is AAMS-CP with $m = 0.35$.

As seen in the experimental results in Figure 5, the best EER is 2.64% in $\beta = 1.4$, which means a 1:1.4 combination of prototypical loss function and margin-based classification loss function is the best setting, namely $L = L_{CP} + 1.4 \cdot L_{AAMS}$. The simple and beautiful form conforms to the Occam's razor principle. The following experiment is conducted with β as 1.4.

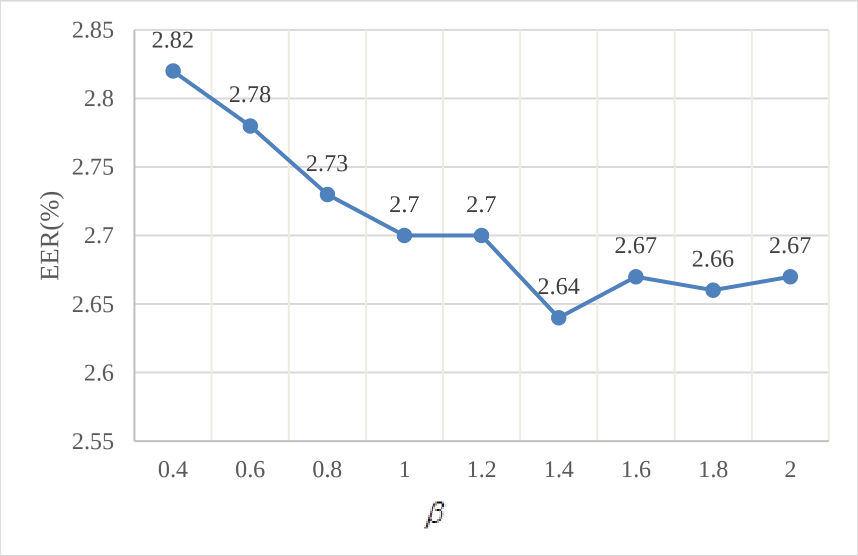

Figure 5. EER in different weight β.

As seen in Table 3, The combination of the proposed two loss functions outperforms any other loss or combination. For example, when the coding layer is NetVLAD, AAM-S+CP outperforms all other loss functions, with an EER of 2.99%, relatively reduced by 48.8%, 19.4%, 6.3%, 19.41% and 13.58% compared with Softmax, CP, Softmax+CP, AM-S and AAM-S, respectively.

Table 3. The comparison of results using different loss function and parameters. For the loss function, CP, AM-S and AAM-S denote Cosine-Prototypical, Additive Margin Softmax and Additive Angular Margin Softmax, respectively. For the neural network, Res refers to fast-ResNet34 network model.

Loss	Res+SP	Res+ASP	Res+NetVLAD	Res+NeXtVLAD	Res+DeltaVLAD
Softmax	5.59	5.65	5.84	5.04	3.99
CP	4.16	4.19	3.71	3.52	2.84
Softmax+CP	3.81	4.16	3.19	3.52	2.78
AM-S	4.05	4.12	3.71	3.34	3.38
AAM-S	4.05	4.08	3.46	3.28	3.37
AM-S+CP (Ours)	**3.52**	**3.47**	3.22	**2.73**	2.73
AAM-S+CP (Ours)	3.69	3.92	**2.99**	2.83	**2.64**

When the coding layer is NeXtVLAD, AM-S+CP outperforms all the other loss functions, with an EER of 2.73%, relatively reduced by 45.8%, 28.9%, 28.9%, 18.3% and 16.8% compared with Softmax, CP, Softmax+CP, AM-S and AAM-S, respectively. Because the AM-S and AAM-S can make the distance between classes larger, and the CP can learn a metric which can make similar samples close and dissimilar ones distant in the embedding space, In addition, when the encoding layer is DeltaVLAD, the EER of AAM-S+CP is 2.64% with a performance improvement of 3.3% compared with AM-S+CP due to its more powerful feature discrimination by maximising classification boundaries in angular space. The effectiveness and superiority of the combination of the AAM-S loss function and the CP loss function have been fully demonstrated in our system with Res+DeltaVLAD.

4.3. Evaluation of DeltaNeXtVLAD

The model with fast-ResNet34, NeXtVLAD and AAM-Softmax is used as baseline, meanwhile, we have also compared several of the better models available today. The Table 4 below gives the results of the comparison.

As seen in Table 4, the result of the proposed encoding layer is 2.64%, clearly superior to the baseline of 3.52%. Besides, we compare several models that work well today in Table 4, the effect of our proposed encoding layers outperforms the effects of all the others. Especially, the model with the DeltaVLAD encoding layer and AAM-S+CP loss function has best performance with an EER of 2.64%, reduced by 10.2%, 18.01%, 15.65% and 24.36% compared with NeXtVLAD, FPM, ANF and SP, respectively. That is because differencing can calculate the dynamic changes between frames, which is more conducive to capturing insignificant changes and NeXtVLAD improves processing accuracy and speed. Meanwhile, the number of parameters of the model with DeltaVLAD is about 1.58M, similar to the others, and no additional parameters are added due to the simplicity of differencing.

From the above results, it can be concluded that our proposed encoding strategy and combined loss function is effective. The result of 2.64% achieves the state-of-the-art in the open-set task. DeltaVLAD's ease of use and simple mathematical theory underline its strengths.

Table 4. Speaker verification results in the standard VoxCeleb1 benchmark. Results of the compared methods are quoted from their original papers.

System	**Encoding layer**	**Loss**	**Parameters**	**dataset**	**EER (%)**
fast-ResNet34 (baseline)	NeXtVLAD	AAM-Softmax	1.58M	Vox1	2.94
ResNet34[33]	FPM	A-Softmax	5.85M	Vox1	3.22
fast-ResNet34[34]	ANF	Proto+Softmax	NR	Vox1	3.13
Modify-ResNet[35]	SP	ParAda-S	1.5M	Vox1	3.49
Ours	NeXtVLAD	AAM-S+CP	1.58M	Vox1	2.83
Ours	DeltaVLAD	AAM-S+CP	1.58M	Vox1	**2.64**

4.4. Ablation studies

Ablative experiments are conducted to verify the proposed DeltaVLAD encoding layer and the combined loss function, and the results are presented in the following Table 5.

Table 5. Ablation studies of loss function and DeltaVLAD strategies.

Module	**Component**			
CP	√		√	√
AAM-S		√	√	√
NeXtVLAD	√	√	√	
DeltaVLAD				√
EER(%)	3.52	3.28	2.83	2.64

As seen in Table 5, several ablation experiments are conducted to verify the random combination of different loss functions and the encoding methods on performance. Separate use of combination CP+NeXtVLAD and combination AAM-S+NeXtVLAD get an EER of 3.52% and 3.28% respectively, and both are higher than that of combination AAM-S+CP+NeXtVLAD, which is 3.18%, but the EER of the combination proposed in this paper is 2.64%, which shows superior performance to the others, which shows the effectiveness and superiority of our proposed approach. That is because the combined loss function has the advantages of both the metric loss function and the classification loss function, which can maximize the inter-class distance and reduce the intra-class distance, and the few-shot feature of CP loss function comes with a class-balanced sampling strategy, which is more suitable for open-set tasks just like speaker recognition. Besides, differential processing of the acoustic spectrum can capture unique changes in sound characteristics between the preceding and following frames, helping to extract more discriminative features.

5. Conclusions

In this paper, we propose an embedding-based speaker recognition model for extracting more discriminative speaker feature vectors in open-set task. The model uses a classical mathematical tool to optimise coding, called DeltaVLAD and a combined loss function, called AAM-S+CP loss function, and experiments conducted verify its effectiveness. DeltaVLAD fusing difference and NeXtVLAD, where difference assists us capture the changes between the preceding and following frames of the current frame and NeXtVLAD encoding method split the frame-level features into multiple word spaces before aggregation, and then perform VLAD operations in each subspace, significantly reducing the number of parameters and improving performance. In addition, we further explored a combined AAM-S+CP loss function, whose clear boundary capabilities reduces the intra-class distance and increases the inter-class distance at the same time, making the features more discriminative. The whole system architecture is simple and generous, in line with Occam's Razor's Law. Experiments conducted on the Voxceleb1 benchmark yield excellent results and demonstrate the effectiveness of the proposed model for the SV task.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61573151 and Grant 6197609, the Science and Technology Planning Project of Guangdong Province under Grant 2018B030323026 and the Young Innovative Talents Project of Guangdong Province under Grant 2018GkQNCX005.

Conflict of interest

The authors declare no conflicts of interest in this paper.

References

1. A. Nagrani, J. S. Chung, A. Zisserman, Voxceleb: A large-scale speaker identification dataset, *Proc. Interspeech*, 2017, 2616–2620. https://doi.org/10.21437/Interspeech.2017-950
2. J. S. Chung, A. Nagrani, A. Zisserman, Voxceleb2: Deep speaker recognition, *Proc. Interspeech*, 2018, 1086–1090. https://doi.org/10.21437/Interspeech.2018-1929
3. D. A. Reynolds, R. Rose, Robust text-independent speaker identification using gaussian mixture speaker models, *IEEE T. Speech Audio Processing*, **3** (1995), 72–83. https://doi.org/10.1109/89.365379
4. T. F. Zheng, L. T. Li, Robustness-related issues in speaker recognition, *Singapore: Springer*, 2017. https://doi.org/10.1007/978-981-10-3238-7
5. D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted gaussian mixture models, *Digit. Signal Process.*, **10** (2000), 19–41. https://doi.org/10.1006/dspr.1999.0361
6. F. Richardson, D. Reynolds, N. Dehak, Deep neural network approaches to speaker and language recognition, *IEEE Signal Proc. Let.*, **22** (2015), 1671–1675. https://doi.org/10.1109/LSP.2015.2420092
7. D. Snyder, D. Garcia-Romero, D. Povey, S. Khudanpur, Deep neural network embeddings for text-independent speaker verification, *Proc. Interspeech*, 2017, 999–1003.

https://doi.org/10.21437/Interspeech.2017-620

8. J. Jung, H. S. Heo, J. Kim, H. Shim, H. J. Yu, RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification, *arXiv*, 2019. Available from: https://arxiv.org/abs/1904.08104.
9. T. Ko, Y. Chen, Q. Li, Prototypical networks for small footprint text-independent speaker verification, In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 6804–6808. https://doi.org/10.1109/ICASSP40776.2020.9054471
10. Y. Liu, L. He, J. Liu, Large margin softmax loss for speaker verification, *Proc. Interspeech*, 2019. 2873–2877. https://doi.org/10.21437/Interspeech.2019-2357
11. Y. F. Wu, C. Guo, H. Gao, X. Hou, J. Xu, Vector-based attentive pooling for text-independent speaker verification, *Proc. Interspeech*, 2020, 936–940. https://doi.org/10.21437/Interspeech.2020-1422
12. S. B. Kim, J. W. Jung, H. J. Shim, J. H. Kim, H. J. Yu, Segment aggregation for short utterances speaker verification using raw waveforms, *Proc. Interspeech*, 2020, 1521–1525. https://doi.org/10.21437/Interspeech.2020-1564
13. C. F. Luo, X. Guo, A. W. Deng, W. Xu, J. H. Zhao, W. X. Kang, Learning discriminative speaker embedding by improving aggregation strategy and loss function for speaker verification, In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*, 2021, 1–8. https://doi.org/10.1109/IJCB52358.2021.9484331
14. E. Variani, X. Lei, E. McDermott, I. L. Moreno, J. Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363
15. B. Desplanques, J. Thienpondt, K. Demuynck, ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification, *Proc. Interspeech*, 2020, 3830–3834. https://doi.org/10.21437/Interspeech.2020-2650
16. A. Senior, H. Sak, I. Shafran, Context dependent phone models for LSTM RNN acoustic modelling, In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, 4585–4589. https://doi.org/10.1109/ICASSP.2015.7178839
17. W. C. Cai, J. K. Chen, M. Li, Exploring the encoding layer and loss function in end-to-end speaker and language recognition system, *arXiv,* 2018. Available from: https://arxiv.org/abs/1804.05160.
18. K. Okabe, T. Koshinaka, K. Shinoda, Attentive statistics pooling for deep speaker embedding, *arXiv*, 2018. Available from: https://arxiv.org/abs/1803.10963.
19. Y. Zhong, R. Arandjelović, A. Zisserman, GhostVLAD for set-based face recognition, In: *Asian conference on computer vision*, Springer, Cham, 2018, 35–50. https://doi.org/10.1007/978-3-030-20890-5_3
20. W. Xie, A. Nagrani, J. S. Chung, A. Zisserman, Utterance-level aggregation for speaker recognition in the wild, In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 5791–5795. https://doi.org/10.1109/ICASSP.2019.8683120
21. R. Lin, J. Xiao, J. Fan, NEXTVLAD: An efficient neural network to aggregate frame-level features for large-scale video classification, In*: Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.
22. C. Zhang, Y. X. Zou, G. Chen, L. Gan, PAN: Towards fast action recognition via learning persistence of appearance. *arXiv*, 2020. Available from: https://arxiv.org/abs/2008.03462.

23. Y. Liu, L. He, J. Liu, Large margin softmax loss for speaker verification, *arXiv*, 2019. Available from: https://arxiv.org/abs/1904.03479.
24. P. Anand, A. K. Singh, S. Srivastava, B. Lall, Few shot speaker recognition using deep neural networks, *arXiv*, 2019. Available from: https://arxiv.org/abs/1904.08775.
25. L. Wan, Q. Wang, A. Papir, I. L. Moreno, Generalized end-to-end loss for speaker verification, In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 4879–4883. https://doi.org/10.1109/ICASSP.2018.8462665
26. S. M. Kye, Y. Jung, H. B. Lee, S. J. Hwang, H. Kim, Meta-learning for short utterance speaker recognition with imbalance length pairs, *arXiv*, 2020. Available from: https://arxiv.org/abs/2004.02863.
27. S. M. Kye, Y. Kwon, J. S. Chung, Cross attentive pooling for speaker verification, In: *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, 294–300. https://doi.org/10.1109/SLT48900.2021.9383565
28. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 7132–7141.
29. J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, et al., In defence of metric learning for speaker recognition, *arXiv*, 2020. Available from: https://arxiv.org/abs/2003.11982.
30. F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification, *IEEE Signal Proc. Let.*, **25** (2018), 926–930. https://doi.org/10.1109/LSP.2018.2822810
31. J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 4690–4699.
32. J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, *arXiv*, 2017. Available from: https://arxiv.org/abs/1703.05175.
33. Y. Jung, S. M. Kye, Y. Choi, M. Jung, H. Kim, Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances, *arXiv*, 2004. Available from: https://arxiv.org/abs/2004.03194.
34. S. M. Kye, J. S. Chung, H. Kim, Supervised attention for speaker recognition, In: *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, 286–293. https://doi.org/10.1109/SLT48900.2021.9383579
35. M. Rybicka, K. Kowalczyk, On parameter adaptation in softmax-based cross-entropy loss for improved convergence speed and accuracy in DNN-based speaker eecognition, *Proc. Interspeech*, 2020, 3805–3809. https://doi.org/10.21437/Interspeech.2020-2264