

Large-scale singer recognition using deep metric learning: an experimental study

Shichao Hu

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

shichaohu@tencent.com

Beici Liang

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

beiciliang@tencent.com

Zhouxuan Chen

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

robinzxchen@tencent.com

Xiao Lu

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

shawlu@tencent.com

Ethan Zhao

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

ethanzhao@tencent.com

Simon Lui

QQ Music BU

Tencent Music Entertainment (TME)

Shenzhen, China

nomislui@tencent.com

Abstract—Singer recognition aims to automatically recognize the singer of a given recording. Compared to spoken voices, singing voice is characterized by a much higher degree of vocal style. The task becomes more challenging when it operates on numerous singers. This paper explores different strategies in a deep metric learning framework, with special focus on their performance in a large-scale dataset consisting of audio samples from 5057 singers. We conduct thorough experiments to compare loss functions, including triplet loss, generalized end-to-end (GE2E) loss, and prototypical network (PN) loss. Effects of vocal source separation is also investigated. Using audio inputs with separated vocals, our model trained with PN loss outperforms other evaluated methods in the identification task. While in the verification task with one-on-one comparison of two single embeddings, triplet loss achieves the best results. However, verification using PN loss shows superior performance to methods with triplet loss when using the centroid of 5 embeddings to represent the singer embedding. Using longer segments for a singer representation consistently improves the performance for all evaluated tasks.

Index Terms—Singer Identification, Singer Verification, Deep Metric Learning, Embedding Model

I. INTRODUCTION

Recognizing the singer identity directly from audio inputs has been a challenging task in Music Information Retrieval (MIR). Compared to speaker recognition from spoken voices, the nature of music recordings makes singer recognition a more difficult task. For instance, background accompaniments could result in a low signal-to-interference ratio, thus disturb singer identification on the vocal parts. Besides, singing voice itself exhibits a more complex characteristics than the spoken voice in time and frequency domains. Besides, some vocal effects (reverb, delay and etc.) of music recordings can further increase the challenges for singing voice recognition. Therefore, singing voice, as a special example of spoken voice, can be used to **test the robustness of current speaker recognition systems**.

Over the past decades, a great number of attempts have been made for speaker recognition based on advanced deep learning methods, which outperforms the conventional statistical signal processing methods such like i-vector [1]. For large-scale speaker recognition, the deep embedding method has been widely used which characterize a speaker with an embedding representation. [2]–[6] The investigation of singer recognition task is mainly inspired by existing speaker recognition systems.

Existing studies for singer recognition attempt to obtain a representation in order to characterize the identity of a singer. One of the earliest attempts applied LPC cepstral coefficients to identify 8 singers using Gaussian Mixture Model (GMM) in [8]. In [9], spectral envelope features are used to perform singer identification on audio samples that contain the singer's voice only. To do the identification on a given music recording, source separation techniques are used to extract singing vocals, which can be represented as i-vector [1] for singer classification in [10]. Recently, the use of deep learning method is investigated such as in [11] which deals with singer identification as a classification task. However, these studies were evaluated on datasets such as *Artist20* [12] with a limited number of singers. This **impedes** the use of deep embedding models to obtain singer embeddings for **generalization purposes**. To address this issue, *JukeBox* dataset including 936 singers [13] was released and used in different methods for singer verification. However, the author claims that the training data is insufficient to train deep embedding models.

With large-scale speech datasets [14]–[16], **metric learning** has been successfully used in speaker recognition on spoken voices. An early attempt in [5] uses triplet loss to train speaker embedding models. A centroid-based metric learning method using prototypical network is proposed in [2] that outperforms the triplet method. Another study in [3] uses generalized end-to-end (GE2E) loss to facilitate the training efficiency for



speaker verification.

However, with much less attention paid to singer recognition study, latest deep embedding models have not yet been fully explored for singer recognition tasks. Besides, current available datasets with a limited number of singers also make it difficult to latest deep learning techniques for singer recognition task.

In order to apply deep embedding models mentioned above to our singer recognition tasks, we create a large-scale dataset consisting of 5057 singers from commercial audio recordings. Based on the same model architecture, this paper focuses on the loss functions as introduced in II. We conduct thorough experiments in Section III to evaluate these functions: 1) in singer identification tasks; 2) in singer verification tasks; 3) when longer segments are used; 4) with or without vocal source separation. Our experiments result in a deep embedding model which is the first system that deals with both singer identification and verification, and achieves strong performance on both tasks.

II. PROPOSED METHOD

Existing deep embedding models address speaker recognition on spoken voices as a classification task [4], [14], or using metric learning [2], [3], [5], [6]. Unlike softmax classification using cross-entropy loss, metric learning optimizes the network by decreasing the inter-class distance while increasing the intra-class separability. In this section, we first present our model architecture and then introduce three loss functions for metric learning.

A. Model Architecture

Inspired by the state-of-the-art models for speaker recognition in [5], [6], we use a model architecture similar to *Thin ResNet34*. Details of our proposed model are presented in Table I. Model configurations are set to accommodate our tasks that use spectrogram as the model input. Self-attentive pooling (SAP) [6], [7] is used to aggregate frame-level features into a segment-level representation. The last fully-connected layer generates the model output with a dimension of 512. We denote the output as embedding x , which is then used for calculating loss with different metric learning strategies.

B. Loss Functions

Here inputs of the following metric learning models are mini-batch segment embeddings with a dimension of $N \times M \times 512$, where the batch size N also refers to the number of singers, and M is the number of embeddings obtained from M audio segments associated with a singer. We set the dimension of embeddings as 512 in the following experimental study.

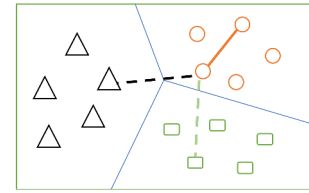
1) *Triplet Loss*: Triplet network is commonly used in the metric learning framework. It was proposed to solve face recognition problems [17] and has been successfully applied to speaker recognition [5].

One triplet sample consists of an anchor x^a , a positive embedding x^+ , and a negative embedding x^- (Fig. 1(a)). The corresponding triplet loss is measured as:

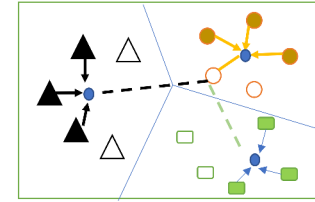
$$L = \max(0, \|x^a - x^+\| - \|x^a - x^-\| + \alpha), \quad (1)$$

TABLE I
THIN RESNET34 ARCHITECTURE

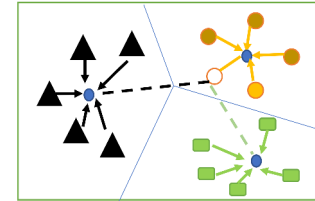
Input Spectrogram ($64 \times 350 \times 1$)	Stride	Output Size
Conv2D (7×7 , 64)	(2,1)	$32 \times 350 \times 16$
Max Pool (3×3) Stride (2×1)	(2,1)	$16 \times 350 \times 16$
$\begin{matrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{matrix}$	$\times 3$ (1,2)	$16 \times 175 \times 16$
$\begin{matrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{matrix}$	$\times 4$ (2,2)	$8 \times 88 \times 32$
$\begin{matrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{matrix}$	$\times 6$ (2,2)	$4 \times 44 \times 64$
$\begin{matrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{matrix}$	$\times 3$ (1,2)	$4 \times 22 \times 128$
Avg Pool (3×1)	(2,2)	$128 \times 22 \times 1$
Self-attentive pooling(SAP)		1×128
FC, 512		1×512



(a) Triplet Loss



(b) PN Loss



(c) GE2E Loss

Fig. 1. Comparison of 3 loss functions for metric learning. Dashed lines represent distances encouraged to increase, while solid lines represent distances being decreased. (a): For Triplet Loss. (b): For Prototypical network (PN) Loss; (c): For GE2E Loss.

where $\alpha > 0$ is a margin parameter. In our experiments, the hard negative mining strategy is adopted. The final loss for a mini-batch is calculated as the averaged loss from all the sampled triplets.

2) *Prototypical Network Loss*: In Prototypical Networks (PN) [2], [18], one mini-batch \mathbf{B} is divided into a support set \mathbf{S} and a query set \mathbf{Q} , namely, $\mathbf{B} = \mathbf{S} \cup \mathbf{Q}$. The number of segment embeddings in the two sets is denoted as S and

Q , respectively, i.e., $M = S + Q$. In our experiments, the query sample is set to the M -th segment of each singer. To calculate the distance d_n between the query sample and the support set regarding the n -th singer, centroid of the support set (also known as *prototype*) is defined as follows:

$$c_n = \frac{1}{|\mathbf{S}_n|} \sum_s x_{s,n} \quad (2)$$

Then Euclidean distance is used to obtain d_n using:

$$d_n = \|x_M - c_n\|^2 \quad (3)$$

A query sample of a singer should differ from the support set of other singers (Fig. 1(b)). This can be formulated as a classification task using softmax over distances to the prototypes of every singers. Finally, the PN loss is calculated as the sum of multi-class cross-entropy of all the query set.

It is noticed that the Euclidean distance in Eq. 3 can be replaced with a cosine-based similarity metric:

$$d_n = w \cdot \cos(x_M, c_n) + b, \quad (4)$$

where w is a scale parameter, and b refers to the bias. Both can be learned in the training process. This is known as the Angular Prototypical Network Loss proposed in [6]. It was validated that using the angular loss function improves the robustness of objective against feature variance and demonstrates more stable convergence [22].

3) *Generalized End-to-End Loss*: Generalized End-to-End (GE2E) loss [3] is similar to the PN loss. The main difference is the selection of the query and support samples. For the positive metric distance of a same singer, every segment embedding in the mini-batch will be compared with the centroids of other segment embeddings of the same singer. The centroids of different classes will be formed by all segments of the corresponding different subject within a mini-batch in order to calculate the negative metric distance (Fig. 1(c)). A similarity matrix is constructed by cosine similarity (defined as Eq. (4)) of each embedding in the mini-batch and all centroids.

III. EXPERIMENTS

In this section, we will introduce the dataset and detail of our experiments. We conduct both the identification and verification task and evaluate our proposed methods with different metric learning methods. As far as our knowledge, our study is the first work to evaluate both singer identification and verification with a large-scale training data.

A. Dataset and Experimental Setup

Our dataset consists of commercial music tracks in MP3 format with a sampling rate of 44.1kHz from 5057 singers. This is comparable to the VoxCeleb2 dataset (5994 speakers) [15] commonly used in speaker recognition tasks. For each singer, we randomly selected 10 tracks from various albums. These tracks were split into 8:2 for training and validation. The testing set includes additional tracks for the evaluation on both “seen” and “unseen” singers. Here we collected another

10 tracks for each 706 singers within the 5057 seen singers, and for each 706 unseen singers.

Every track in our dataset was firstly downsampled to 16kHz and then processed into a mel-spectrogram with 64 bins using a hamming window of 25 ms and a hop size of 10ms. With the help of lyric information aligned to the audio, time stamps of onsets and offsets of each phrase can be obtained. Thus clips with vocals were selected according to these time stamps. In the end, 3.5-second mel-spectrograms were used as inputs to train the proposed models. It corresponds to a feature with a dimension of 350×64 . To explore the effect of vocal source separation, we also used mel-spectrograms from vocal-only clips, which were separated from the original clips using *Spleeter* [19].

Our implementation is based on PyTorch [20] and trained for 500 epochs with a batch size of $N = 400$ for each training task. The Adam optimizer is used with an initial learning rate of 0.001, which decreases by 5% every 10 epochs. In terms of the hyper-parameters for metric learning, we set α to 0.1 or 0.2 for triplet loss. For training with GE2E and prototypical networks, the number of singers M is set to 3 or 4.

B. Evaluation on Identification Tasks


In terms of the identification evaluation for the selected 706 singers in both “seen” and “unseen” case, the task here is to choose one of them by calculating the distance between the query feature embedding and all candidate singer embeddings. For the 10 selected songs of each singer, the centroid of embedding features from 6 songs are calculated in order to represent the prototype of each singer. A single segment embedding (3.5s) randomly selected from the remaining 4 songs is evaluated as the query input. Since the identification evaluation is conducted for every one of the 706 singers, the final accuracy of an evaluated method is calculated as the average value of all the identification trails.

The results for “seen” and “unseen” testing, with and without source separation, are presented in Table II. It suggests that the prototypical networks (PNL and APNL) achieve the best results in general, with a significantly superior performance than the popular triplet loss used in speaker recognition systems [5], [21] and a recent singer recognition study [13]. Moreover, the source separation benefits identification performance for all evaluated methods.

C. Evaluation on One-on-One Verification Tasks

The verification task is conducted by a one-on-one comparison of a embedding pair from respectively two singers, with 120 iterations for each singer of the “seen” and “unseen” singers. The performance is evaluated by equal error rate (EER). The result (right column of Table II) shows that triplet loss outperforms other loss functions including the prototypical network. This seems to conflict with the finding in the identification task. It may be explained by that the one-on-one comparison of the embedding pair in the verification task exactly matches the scenario of the triplet training, which optimizes the distance from one embedding (anchor) to the

TABLE II
IDENTIFICATION (1 VS 706) AND VERIFICATION FOR SEEN AND UNSEEN SINGERS UNDER ONE EMBEDDING (3.5s) QUERY USING MODELS TRAINED WITH TRIplet LOSS(TRIPLETS), GENERALISED END-TO-END LOSS(GE2E), PROTOTYPICAL NETWORK LOSS(PNL), AND ANGULAR PROTOTYPICAL NETWORK LOSS(A-PNL).



Training Metrics	Identification				Verification	
	Top-1 (%)		Top-5 (%)		EER (%)	
	Seen	Unseen	Seen	Unseen	Seen	Unseen
With source separation						
Triplets ($\alpha=0.1$)	42.53	40.66	69.26	62.95	16.32	20.34
Triplets ($\alpha=0.2$)	43.48	38.93	69.69	64.54	16.36	20.11
GE2E (M=3)	50.99	45.59	72.80	67.58	16.44	21.42
GE2E (M=4)	53.25	44.43	74.78	67.87	17.09	21.78
PNL (M=3)	51.56	43.56	74.50	66.71	17.75	20.97
PNL (M=4)	53.68	47.46	74.93	69.46	20.34	23.55
A-PNL (M=3)	51.27	45.59	73.65	68.60	16.67	20.62
A-PNL (M=4)	55.81	49.35	76.34	70.77	17.06	20.64
Without source separation						
Triplets ($\alpha=0.1$)	40.08	35.46	64.59	59.62	15.87	18.97
Triplets ($\alpha=0.2$)	43.48	35.02	65.01	63.10	15.92	19.10
GE2E (M=3)	47.73	38.35	69.83	2.52	16.58	20.57
GE2E (M=4)	47.59	42.11	72.38	65.41	17.37	21.35
PNL (M=3)	47.73	42.98	70.82	68.74	17.58	20.75
PNL (M=4)	51.98	44.57	72.66	68.31	20.04	22.50
A-PNL (M=3)	50.00	43.85	71.10	67.87	16.10	19.66
A-PNL (M=4)	49.29	43.41	70.96	66.86	17.14	20.03

other (positive or negative). However, the identification task defines a singer representation as the centroid of all segment embeddings for building the searching library. This is the same as the scenario of prototypical network in terms of the prototype calculation. To validate this, we conduct a following experiment to test the effects of multiple segments for generating a representative embedding for both the identification and verification task.

Another interesting finding in the singer verification task is that the performance is worse with source separation than without. This is probably because that large-scale training of music data improves the robustness for singer representation of a short segment even with the presence of music background.

D. Evaluation using longer segments

Since singing voice exhibits a broad dynamics of vocal styles, embeddings of multiple segments may help generate more stabilized representation of the target singer. To validate this, we test the identification task with multiple segments as input. The query embedding is calculated as the centroid of multiple segment embeddings. The result is shown in Table III.

TABLE III
IDENTIFICATION TEST WITH VARIOUS NUMBER OF SEGMENTS AS QUERY INPUT FOR THE SINGER IDENTIFICATION TASK (TOP-1 RESULT) USING MODELS TRAINED WITH TRIplet LOSS(TRIPLETS), AND ANGULAR PROTOTYPICAL NETWORK LOSS(A-PNL).

No. of seg.	A-PNL, $M=4$		TPL, $\alpha=0.2$	
	Seen	Unseen	Seen	Unseen
With source separation, Top-1 (%)				
1	55.81	49.35	43.48	38.93
3	86.40	77.13	78.61	67.87
5	93.34	87.55	86.96	83.07
Without source separation, Top-1 (%)				
1	49.29	43.42	40.93	35.02
3	80.59	67.73	73.51	67.73
5	93.06	84.95	87.96	81.19

It shows that testing with more segment embeddings can significantly improve the identification performance consistently in all evaluated methods.

It should be noted that a centroid of multiple segment embeddings is an appropriate option if not the best to be

TABLE IV
VERIFICATION TEST WITH VARIOUS LENGTH OF SEGMENTS FOR CALCULATING SINGER PROTOTYPES USING MODELS TRAINED WITH TRIPLET LOSS (TRIPLETS), AND ANGULAR PROTOTYPICAL NETWORK LOSS(A-PNL).

Metrics	1 Seg.		3 Seg.		5 Seg.	
	Seen	Unseen	Seen	Unseen	Seen	Unseen
With source separation, EER(%)						
Triplets ($\alpha=0.1$)	16.32	20.34	10.46	13.59	9.14	12.11
Triplets ($\alpha=0.2$)	16.36	20.11	10.46	13.38	9.21	11.66
A-PNL (M=3)	16.67	20.62	10.50	13.75	9.24	11.93
A-PNL (M=4)	17.06	20.64	10.58	13.59	9.07	11.62
Without source separation, EER(%)						
Triplets ($\alpha=0.1$)	15.87	18.97	10.34	12.72	9.16	11.21
Triplets ($\alpha=0.2$)	15.92	19.10	10.36	12.68	8.98	11.04
A-PNL (M=3)	16.10	19.66	10.18	12.79	8.72	10.99
A-PNL (M=4)	17.14	20.03	10.66	13.12	9.15	11.11

determined as a representative embedding of a singer, since it improves recognition performances for all evaluated methods including training with triplet loss. **Taking into account that singing voices have much higher variability compared with spoken voices, a single or a centroid embedding may not be very enough to represent a singer.** This will be investigated as one of our future works.

The verification task here investigates the effects of **longer segments for embedding representation**. This is different from the one-on-one verification in Table II where two single embeddings are compared. Here, **we select a certain number of embedding segments ($s = 1, 3, 5$) to calculate their centroid as a representative embedding**. For verification, the decision is made by comparing the distance of a selected query embedding and the representative embedding of a singer. Table IV shows the result. The verification performance consistently improves with centroid embedding from longer segments for singer representation. Triplet loss performs better for verification task with one single embedding for enrollment. However, the A-PNL shows better advantage with more segments ($s = 5$) for the prototype representation. It can be explained by that the one-on-one verification matches the triple training method. While calculating a centroid of multiple segment embeddings is consistent the prototype calculation of prototypical networks.

IV. CONCLUSIONS

This paper tackles the singer recognition problems. We conduct a thorough experimental study based on a large-scale singer training data using different metric learning functions. We evaluate the performance on both singer identification and verification tasks. The results suggest that the prototypical network with vocal source separation performs best for the identification task. While for one-on-one singer verification task, triplet loss without source separation achieves the best

result. The centroid embedding for a singer representation using longer segments greatly improves the performance for both tasks. When using 3 or 5 embedding segments to represent the singer prototype in the verification task, prototypical networks start to outperform the triplet loss. The results may be explained by that the one-on-one verification task exactly matches the scenario of triplet training (distance from an anchor embedding to a positive or negative embedding); Singer representation by the centroid of multiple segments in the identification and longer-segments verification task is consistent with the prototype calculation in prototypical networks. Considering the high variability in the singing voice, a single or a centroid embedding may not be enough to represent a singer. This will be discussed as a future work of the present study.

REFERENCES

- [1] Najim Dehak, Patrick J Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2010.
- [2] Jixuan Wang, Kuan-Chieh Wang, Marc T Law, Frank Rudzicz, and Michael Brudno, "Centroid-based deep metric learning for speaker recognition," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 3652–3656.
- [3] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 4879–4883.
- [4] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Senior, "Utterance-level aggregation for speaker recognition in the wild," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 5791–5795.
- [5] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," arXiv preprint arXiv:1705.02304, vol. 650, 2017.
- [6] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and

- Icksang Han, "In Defence of Metric Learning for Speaker Recognition." *Proc. Interspeech 2020* (2020): 2977-2981.
- [7] Weicheng Cai, Jinkun Chen, and Ming Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*. 2018.
 - [8] Tong Zhang, "Automatic singer identification," in 2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698). IEEE, 2003, vol. 1, pp. 1-33.
 - [9] Mark A Bartsch and Gregory H Wakefield, "Singin voice identification using spectral envelope estimation," *IEEE Transactions on speech and audio processing*, vol. 12, no. 2, pp. 100-109, 2004.
 - [10] Sharma, Bidisha, Rohan Kumar Das, and Haizhou Li. "On the Importance of Audio-Source Separation for Singer Identification in Polyphonic Music." In *INTERSPEECH*, pp. 2020-2024. 2019.
 - [11] Zain Nasrullah and Yue Zhao, "Music artist classification with convolutional recurrent neural networks," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1-8.
 - [12] Daniel PW Ellis, "Classifying music audio with timbral and chroma features," 2007.
 - [13] Anurag Chowdhury, Austin Cozzo, and Arun Ross, "Jukebox: A multi-lingual singer recognition dataset," *Proc. Interspeech 2020*: 2267-2271.
 - [14] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *Proc. Interspeech*, 2017, pp. 2616-2620.
 - [15] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. Interspeech*, 2018, pp. 1086-1090.
 - [16] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech Language*, vol. 60, pp. 101027, 2020.
 - [17] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman, "Deep face recognition," 2015.
 - [18] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077- 4087.
 - [19] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, pp. 2154, 2020.
 - [20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026-8037.
 - [21] Anurag Chowdhury and Arun Ross, "Fusing MFCC and LPC features using 1D triplet CNN for speaker recognition in severely degraded audio signals," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1616-1629, 2019.
 - [22] Wang, Jian, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. "Deep metric learning with angular loss." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593-2601. 2017.