



# Deep Speaker Embedding with Frame-Constrained Training Strategy for Speaker Verification

*Bin Gu*

National Engineering Laboratory for Speech and Language Information Processing,  
University of Science and Technology of China, Hefei, China

bin2801@mail.ustc.edu.cn

## Abstract

Speech signals contain a lot of side information (content, stress, etc.), besides the voiceprint statistics. The session-variability poses a huge challenge for modeling speaker characteristics. To alleviate this problem, we propose a novel frame-constrained training (FCT) strategy in this paper. It enhances the speaker information in frame-level layers for better embedding extraction. More precisely, a similarity matrix is calculated based on the frame-level features among each batch of the training samples, and a FCT loss is obtained through this similarity matrix. Finally, the speaker embedding network is trained by the combination of the FCT loss and the speaker classification loss. Experiments are performed on the VoxCeleb1 and VOiCES databases. The results demonstrate that the proposed training strategy boosts the system performance.

**Index Terms:** Speaker verification, loss function, local variation, frame-level features

## 1. Introduction

Speaker verification (SV) is the task of determining whether a given speech utterance belongs to a specific speaker. In recent years, more attention has been paid to the deep neural network (DNN) for modeling speaker characteristics. Several DNN architectures, including the time delay neural network (TDNN) [1], the convolutional neural network (CNN) [2], and the long short-term memory network (LSTM) [3], were used for speaker representation learning. Their subsequent improvements have consistently achieved high performance gains [4–6] compared with the conventional methods.

In most DNN based systems, the networks are trained to discriminate between speakers with softmax loss. However, this loss does not have a constraint on minimizing the intra-class variance, which might lead to suboptimal results. A possible solution consists in using some auxiliary loss functions. For instance, the center loss [7, 8] was introduced to minimize the distance between each sample and the corresponding class center. On the other hand, the enhanced variants of softmax loss, including A-softmax [9], AM-softmax [10] and AAM-softmax [11], became popular for their improved performance. These methods increase the inter-class variance by introducing a margin penalty to the target logit, and thus they also reduce the intra-class variance. Furthermore, the metric learning methods [12–16], such as the triplet loss [12] and the pairwise loss [13–15], were used in several SV tasks. In metric learning, the training losses are directly computed from the comparison of distances between training instances.

Though above-mentioned methods have achieved a great

success on various x-vectors systems, most of them can still be improved. In general, a deep speaker embedding system could be divided into frame-level, pooling, and utterance-level layers. Above-mentioned works are usually applied at utterance-level layers, which eliminate the inter-session variability to some extent, but the influence of intra-session variability is not considered. In fact, frame-level features contain a lot of side-information such as content and stress [17]. The intra-session variability cause that some frames of an utterances are less speaker-discriminative [18], resulting in less discriminative speaker embeddings. Based on the situation, there comes an intuitive idea to explicitly model the constraint to the frame-level features, in order to enhance the representative power of the local features. In this paper, a frame-constrained training (FCT) strategy is proposed to make frame-level features more speaker-discriminative, and the strategy acts as an auxiliary task in model training. Specifically, the outputs of the frame-level feature extractor are considered as independent speaker representations, while a similarity matrix is calculated among each batch of the training samples. A carefully designed FCT loss is then obtained to optimize the network combined with the speaker classification loss.

To evaluate the effectiveness of the proposed methods, we conducted experiments on the VoxCeleb1 and VOiCES datasets. The experimental results show that the proposed method can boost the discriminative power for speaker embedding extraction, and therefore improve the system performance.

The remainder of this paper is organized as follows. Section 2 introduces the deep speaker embedding model. Section 3 details the proposed method in detail. The experimental setup, the results and the corresponding analysis are presented in section 4. Finally, the conclusions are given in section 5.

## 2. Deep Speaker Embedding Model

A typical speaker embedding network is composed of frame-level, pooling, and utterance-level layers. The frame-level feature extractor of our baseline system is adapted from the backbone ResNet-34 architecture. It comprises an input convolutional layer and four residual stages. Each residual stage includes a set of residual blocks having the same resolution, where each block is composed of two convolutional layers. A statistic pooling layer is employed to convert the variable-length frame-level representations into a fixed-length vector. The two embedding layers are incorporated in the utterance-level layers to extract the speaker embeddings. The network is finally optimized with the AM-softmax loss function. Details about the network architecture are shown in Table 1.

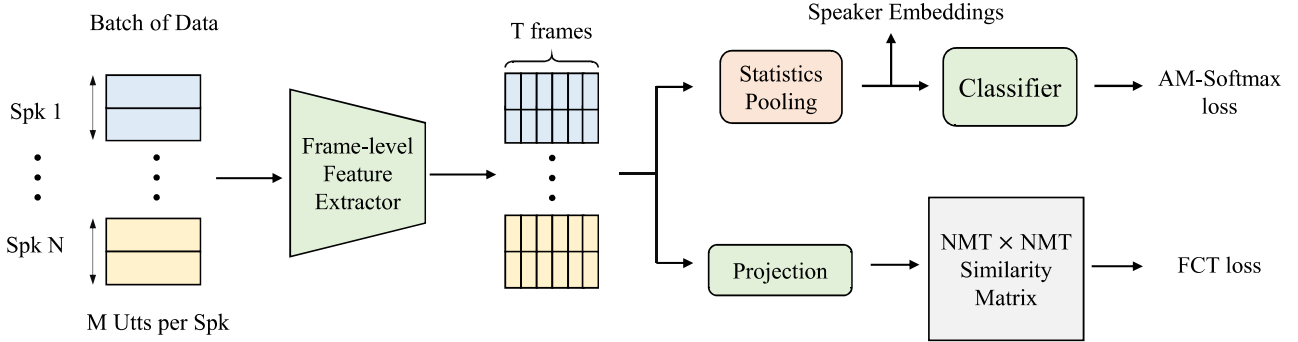


Figure 1: Overview of the proposed system.

Table 1: The ResNet34 based backbone network configuration, where the matrices represent the shape of residual blocks and the multipliers of matrices are the number of stacked blocks. The input size is  $T \times 64 \times 1$ .

Stages	Layer name	Structure	Output size
-	Conv2D-1	$7 \times 7, 32$	$T/2 \times 32 \times 32$
1	ResBlock-1-x	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	$T/2 \times 32 \times 32$
2	ResBlock-2-x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$	$T/2 \times 16 \times 64$
3	ResBlock-3-x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$	$T/2 \times 8 \times 128$
4	ResBlock-4-x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$	$T/2 \times 4 \times 256$
-	StatsPooling	Mean and Std	2048
-	Embedding layer	Dense	512
-	-	Dense	512
-	Output layer	AM-Softmax	$N_{\text{spk}}$

### 3. Frame-Constrained Training Strategy

Fig.1 shows a schematic representation of the frame-constrained training strategy. The proposed method is based on processing a number of utterances at once, in the form of a mini-batch. The frame-level speaker representations are used to obtain the FCT loss function, while a multi-task learning strategy is used to incorporate the FCT loss into the conventional classification loss function.

#### 3.1. Frame-constrained training loss function

We assume that  $N \times M$  equal-length utterances from  $N$  different speakers exist in a mini-batch, where each person has  $M$  utterances. For an utterance  $x_{n,m}$ , the corresponding  $T$  frame-level features can be expressed as:

$$\{e_{n,m}^1, \dots, e_{n,m}^T\} = f_{\theta}(x_{n,m}) \quad (1)$$

where  $f_{\theta}(\cdot)$  is the feature extractor with parameter  $\theta$ . To better adapt these features into a shared metric space, each one is concatenated with the standard deviation  $\sigma_{n,m}$  and projected into a linear space:

$$\mu_{n,m} = \frac{1}{T} \sum_{t=1}^T e_{n,m}^t \quad (2)$$

$$\sigma_{n,m}^2 = \frac{1}{T} \sum_{t=1}^T (e_{n,m}^t - \mu_{n,m})^2 \quad (3)$$

$$g_{n,m}^t = f_w(e_{n,m}^t, \sigma_{n,m}) \quad (4)$$

where  $f_w(\cdot)$  denotes a fully-connected layer with trainable parameter matrix  $W$ . It is worth noting that the utterance-level speaker embedding extraction process is similar to Eq.(4), except that the former uses  $\mu_{n,m}$  instead of  $e_{n,m}^t$ . Therefore the generated  $g_{n,m}^t$  can be considered as fine-grained speaker embedding which contains more local information, compared with its utterance-level counterpart. For the sake of clarity, a generated vector  $g_{n,m}^t$  is considered as frame-based speaker embedding (FSE) referred to as  $g_i (1 \leq i \leq K, K \triangleq N \times M \times T)$ , while the corresponding speaker labels are denoted by  $y_i$ .

Based on the FSEs, a similarity matrix is calculated and the FCT loss is given by:

$$\ell_{FC} = \frac{1}{K \times K} \sum_{i,j} d_{i,j} \quad (5)$$

$$d_{i,j} = \begin{cases} \max(0, \|g_i - g_j\|_2 - \alpha_i), & y_i = y_j \\ -\min(0, \|g_i - g_j\|_2 - \beta_i), & y_i \neq y_j \end{cases} \quad (6)$$

where  $\|\cdot\|_2$  represents the Euclidean norm and  $d_{i,j} (1 \leq i \leq K$  and  $1 \leq j \leq K)$  is the similarity matrix element.  $\alpha_i$  and  $\beta_i$  are respectively two scales limiting the boundaries of the intra-class and inter-class samples corresponding to  $g_i$ . Those intra-class samples should be driven closer. However, over strict constraint may result in information loss, which has shown to be useful for modeling speaker characteristics [19]. Therefore, we just constrain those positive speaker pairs whose distance exceeds  $\alpha_i$ . Finally, the AM-softmax loss and the FCT loss are jointly optimized, and the total loss is given by:

$$\ell_{total} = \ell_{AM} + \lambda \ell_{FC} \quad (7)$$

where  $\lambda$  is a weight factor. Noting that the auxiliary branch for calculating FCT loss will be discarded during inference, and speaker embeddings are still extracted through a standard statistics pooling layer, which does not introduce extra parameters to the model.

Among numerous frame-level negative pairs, only a part of them is selected to calculate the final loss according to the decision boundary in Eq.(6). Intuitively, two frames in a selected negative pair contains similar local contextual information, so that they are more confusing in a SV system. Since the intra-session variability in selected two frames could be neglected, driving the two frames away from each other might separate speaker information from the intra-session variability. In addition, previous methods [9–15] are used at utterance-level, which is different from our work. Since the proposed learning strategy gives a model a more detailed optimization target at frame-level, and therefore the FCT loss could serve as a good supplement to existing works.

### 3.2. Margin types

It can be deduced from the Eq.(6) that the calculation of the FCT loss involves the margins,  $\alpha_i$  and  $\beta_i$ . Two types of methods are then explored to select the values of these key parameters:

- Static method: All the margins are manually set before the training process:

$$\begin{aligned}\alpha_i &= \alpha \\ \beta_i &= \beta\end{aligned}\quad (8)$$

Note that  $\alpha$  and  $\beta$  are two global fixed values. This method is straightforward, and it allows to investigate the system performance under different margin values.

- Dynamic method: The dynamic margins are adaptively set according to the data distribution in a mini-batch :

$$\begin{aligned}\alpha_i &= \frac{1}{\sum_{j=1}^K \delta(y_i=y_j)} \sum_{j=1}^K \delta(y_i=y_j) \cdot \|g_i - g_j\|_2 \\ \beta_i &= \frac{1}{\sum_{j=1}^K \delta(y_i \neq y_j)} \sum_{j=1}^K \delta(y_i \neq y_j) \cdot \|g_i - g_j\|_2\end{aligned}\quad (9)$$

where  $\delta(\cdot)$  is an indicator function. Note that the dynamic method avoids the influence of the extra hyper-parameters to the system, and make the model easier to train.

## 4. Experiments

### 4.1. Datasets and evaluation metrics

The performance of the proposed method is evaluated on the test portion of the VoxCeleb1 and VOiCES datasets. VoxCeleb1 is a standard database used to test SV performance in noisy conditions [20]. Due to the fact that they are sampled from the real world, these datasets include background noise such as laughter and music. The VOiCES dataset, used for speaker verification, is described in the ‘‘VOiCES from a Distance Challenge 2019’’ [21], where the development portion is used to tune the hyper-parameters. The speech samples are recorded with distinct reverberation conditions and noise distractions, which makes the SV task more challenging. The training dataset includes the development portion of both VoxCeleb1 and VoxCeleb2 [22]. In

order to improve the robustness of the systems, the 4-fold augmentation strategy which combines clean data with three copies of augmented data, is performed. More precisely, the data augmentation procedure in Kaldi [23] is followed.

The results are reported in terms of the equal error rate (EER) and the minimum of the normalized detection cost function (minDCF), with the prior target probability  $P_{tar}$  set to 0.01.

### 4.2. Implementation details

The 64-dimensional log-mel filter-bank energies (Fbank) features are used as acoustic features. The mean normalization over a 3 s sliding window, is used. The energy-based voice activity detection (VAD) is used to filter out non-speech frames. Feature processing is implemented using the Kaldi toolkit. The acoustic features are then randomly cropped to lengths of 2-4 s. 64 utterances with the same duration are grouped into a batch. Note that no special sampling strategy is used to construct a batch of training data, and the values of  $N$  and  $M$  described in section 3 are not specifically assigned. The speakers are uniformly sampled in each batch to ensure a uniform distribution of speakers across training, while the utterances are randomly sampled from these speakers. Consequently, the number of utterances from different speakers in a batch may be unequal.

All the models are built using TensorFlow, and optimized using an Adam optimizer. The AM-softmax loss is used with hyper-parameters  $s$  and  $m$  respectively set to 30 and 0.15. Unless otherwise specified, all the experiments setups are similar to those of the baseline system. Other configurations of each system are summarized as:

**Baseline:** This consists of a deep embedding learning baseline system which is based on the Res-Net34 architecture. To prevent overfitting, we use the same type of L2 weight decay and batch normalization as in [24].

**FCT-FM:** The FCT loss and the fixed margin are used in this system.  $\alpha$  and  $\beta$  are set to 0.1 and 1.0, respectively. The weight factor  $\lambda$  is set to 0.1.

**FCT-DM:** The dynamic margin is used to obtain the FCT loss, and  $\lambda$  is set to 0.001.

The two proposed systems project the features into a 512-dimensional metric space, for calculating the similarity. The development set is mainly used to select the hyper-parameters that can guarantee the convergence of the model.

Length normalization, centering and LDA are applied in order to reduce the feature dimension, and enhance the speaker discrimination. The PLDA model training and PLDA scoring are then performed using Kaldi.

### 4.3. Main results

Table 1 reports the comparison results of the baseline and the proposed systems. It can be observed that the FCT loss highly improves the SV performance on both test sets. When no data augment is applied, the FCT-FM system achieves the best results for the VoxCeleb1 dataset, and the EER is reduced from 1.95% to 1.68%. As for the VOiCES datasets, the FCT-DM system achieves improvements over the baseline by approximately 10% in terms of EER and minDCF. Moreover, the FCT-DM achieves the best result on both sets when the training data is enriched. This demonstrates the effectiveness of the proposed methods. Although FCT loss can boost the system performance regardless of which kind of margin is used, there are

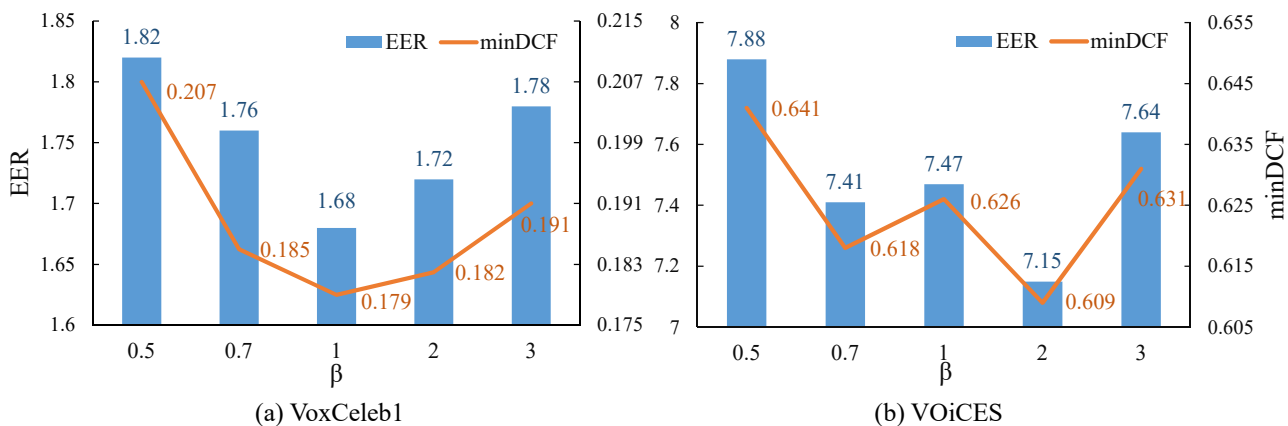


Figure 2: FCT-FM system results on the VoxCeleb1 (a) and the VOICES (b) datasets with different  $\beta$  values.

Table 2: Results of different systems using the VoxCeleb1 and VOICES datasets. “Aug” denotes the data augmentation.

Systems	Aug	VoxCeleb1		VOICES	
		EER	minDCF	EER	minDCF
Baseline	No	1.95	0.215	8.30	0.651
FCT-FM	No	1.68	0.179	7.47	0.626
FCT-DM	No	1.76	0.192	7.37	0.592
Baseline	Yes	1.57	0.160	5.75	0.393
FCT-FM	Yes	1.51	0.154	5.28	0.381
FCT-DM	Yes	1.49	0.147	5.26	0.377

more hyper-parameters in the FCT-FM system, and therefore FCT-DM can be more easily implemented than FCT-FM.

#### 4.4. Further analysis

In this section, a set of experiments is conducted to analyze the influence of different system configurations, on the obtained results. The data augment technology is not used, in order to make the analysis more intuitive.

Firstly, we assess the effect of the fixed margins in Eq.(8), while the experiments are based on the FCT-FM system. Due to the fact that we are more interested in the inter-class distance, we fix  $\alpha$  to 0.1, while varying the value of  $\beta$  from 0.5 to 3.0. The results of the two test sets are shown in Fig. 2. The best performances are obtained for  $\beta$  equals 1 and 2 with VoxCeleb1 and VOICES, respectively. The system performances are adversely affected with other margins.

In the following experiments, we analyze the effect of the projection dimension of the metric space in Eq.(4). It can be seen from Table 2 that a small performance fluctuation exists with different setups. This indicates that the proposed method is not sensitive to the dimension of the metric space.

The Euclidean distance of two samples is calculated in Eq.(6). An evaluation of the system performance using different similarity metrics, is shown in Table 3. It can be seen that the system using the Euclidean distance outperforms the one using the Cosine similarity. This is due to the fact that the former considers both the angle and amplitude information that are crucial

Table 3: FCT-DM system results with different projection dimension.

Dimension	VoxCeleb1		VOICES	
	EER	minDCF	EER	minDCF
1024	1.79	0.197	7.37	0.609
512	1.76	0.192	7.37	0.592
256	1.84	0.197	7.45	0.581

Table 4: FCT-DM system results with different types of similarity metrics.

Metrics	VoxCeleb1		VOICES	
	EER	minDCF	EER	minDCF
Cosine	1.83	0.202	7.63	0.617
Euclidean	1.76	0.192	7.37	0.592

in frame-level speaker embeddings.

## 5. Conclusions

This paper proposed the frame-constrained training (FCT) strategy to reduce speaker-irrelative variability. FCT makes the frame-level features discriminative, and boosts the power of the speaker embeddings. Although the auxiliary branch for obtaining FCT loss introduces extra parameters, and increases the computation cost during the training process, it will be discarded during inference. In practice, there are no additional costs when extracting speaker embeddings, compared with the baseline system. Finally, the experimental results confirm that the proposed approach is beneficial and efficient in deep speaker embedding learning.

## 6. Acknowledgements

This work was partially funded by the National Natural Science Foundation of China (Grant No. U1836219).

## 7. References

- [1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.
- [2] Z. Gao, Y. Song, I. V. McLoughlin, W. Guo, and L.-R. Dai, "An improved deep embedding learning method for short duration speaker verification," in *Interspeech*, 2018, pp. 3578–3582.
- [3] C.-P. Chen, S.-Y. Zhang, C.-T. Yeh, J.-C. Wang, T. Wang, and C.-L. Huang, "Speaker characterization using tdnn-lstm based speaker embedding," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6211–6215.
- [4] R. Zhang, J. Wei, W. Lu, L. Wang, M. Liu, L. Zhang, J. Jin, and J. Xu, "Aret: Aggregated residual extended time-delay neural networks for speaker verification," in *Interspeech*, 2020, pp. 946–950.
- [5] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system," *system*, vol. 13, no. 15, p. 17, 2019.
- [6] S. Yadav and A. Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6794–6798.
- [7] N. Li, D. Tuo, D. Su, Z. Li, D. Yu, and A. Tencent, "Deep discriminative embeddings for duration robust speaker verification," in *Interspeech*, 2018, pp. 2262–2266.
- [8] S. Wang, Z. Huang, Y. Qian, and K. Yu, "Discriminative neural embedding learning for short-duration text-independent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1686–1696, 2019.
- [9] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [10] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.
- [12] A. Jati, R. Peri, M. Pal, T. J. Park, N. Kumar, R. Travadi, P. G. Georgiou, and S. Narayanan, "Multi-task discriminative training of hybrid dnn-tvm model for speaker verification with noisy and far-field speech," in *Interspeech*, 2019, pp. 2463–2467.
- [13] 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.
- [14] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.
- [15] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. 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.
- [16] H. Zhang, L. Wang, K. A. Lee, M. Liu, J. Dang, and H. Chen, "Meta-learning for cross-channel speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5839–5843.
- [17] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.
- [18] M. Ajili, J.-F. Bonastre, W. B. Kheder, S. Rossato, and J. Kahn, "Phonological content impact on wrongful convictions in forensic voice comparison context," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2147–2151.
- [19] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. Cernocký, "On the usage of phonetic information for text-independent speaker embedding extraction," in *Interspeech*, 2019, pp. 1148–1152.
- [20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.
- [21] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828*, 2019.
- [22] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.
- [23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.
- [24] H. Zeinali, L. Burget, J. Rohdin, T. Stafylakis, and J. Cernocký, "How to improve your speaker embeddings extractor in generic toolkits," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6141–6145.