Speaker Verification in Multi-Speaker Environments Using Temporal Feature Fusion

Ahmad Aloradi, Wolfgang Mack, Mohamed Elminshawi, Emanuël A. P. Habets
International Audio Laboratories Erlangen*, Am Wolfsmantel 33, 91058 Erlangen, Germany
{ahmad.aloradi,wolgang.mack,mohamed.elminshawi,emanuel.habets}@audiolabs-erlangen.de

***Abstract*—Verifying the identity of a speaker is crucial in modern human-machine interfaces, e.g., to ensure privacy protection or to enable biometric authentication. Classical speaker verification (SV) approaches estimate a fixed-dimensional embedding from a speech utterance that encodes the speaker's voice characteristics. A speaker is verified if his/her voice embedding is sufficiently similar to the embedding of the claimed speaker. However, such approaches assume that only a single speaker exists in the input. The presence of concurrent speakers is likely to have detrimental effects on the performance. To address SV in a multi-speaker environment, we propose an end-to-end deep learning-based SV system that detects whether the target speaker exists within an input or not. First, an embedding is estimated from a reference utterance to represent the target's characteristics. Second, frame-level features are estimated from the input mixture. The reference embedding is then fused frame-wise with the mixture's features to allow distinguishing the target from other speakers on a frame basis. Finally, the fused features are used to predict whether the target speaker is active in the speech segment or not. Experimental evaluation shows that the proposed method outperforms the x-vector in multi-speaker conditions.**

***Index Terms*—robust speaker verification, speaker detection, multi-speaker speakers, noisy environment**

I. Introduction

Speaker verification (SV) is a binary classification task that aims at verifying the claimed identity of a speaker. The pipeline of classical SV systems consists of a front-end that projects speech onto speaker-discriminative embeddings followed by a back-end to compute the verification scores. Training speaker-discriminative embeddings is commonly done by optimizing classification or metric learning losses. In the metric learning approach, embeddings are trained to capture the similarities between utterances of the same or different speakers by optimizing a metric learning objective, e.g., the triplet loss [1] or the contrastive loss [2]. In the classification approach, the system is first trained to capture the speakers' characteristics by classifying speakers in the training set [3, 4]. At inference time, generalization to unseen speakers is achieved by adopting bottleneck features as embeddings.

The most prominent baseline of the classification-based systems is the x-vector system, which can be currently considered state-of-the-art in SV [5, 6]. The rising popularity of the x-vector inspired various works to improve upon its architecture and training procedure. For example, the frame-level feature extraction was advanced by including elements of the ResNet architecture [7]. The statistics pooling, which aggregates the frame-level mean and standard deviation, was also refined by incorporating different attention mechanisms [8, 9, 10]. Moreover, the vanilla softmax loss was replaced by more sophisticated classification losses that yielded improved performance [11].

Most research in SV assumes that there is only one speaker per input segment, which is reflected by the majority of the standard evaluation datasets [12]. However, real-world recordings might violate the premise of a single speaker. The abundance of single-speaker datasets encouraged embracing this assumption and developing systems accordingly. Nowadays, state-of-the-art SV generally functions by capturing the speaker's characteristics in fixed-dimensional embeddings. If there are multiple speakers to represent, the effectiveness of the learned embeddings is reduced [12]. In such scenarios, it is common to use speaker diarization (i.e., to identify who spoke when [13]) to improve the verification performance [10, 12, 14].

However, diarization is better suited in scenarios for which speakers are only partially overlapping. The emphasis in the present study is on significant overlaps between speakers, which may generally resemble mixtures in speech separation. In such conditions, a speaker is verified if his or her speech is present in a segment containing speech from multiple speakers [15].

We hypothesize that answering the question "does the target speaker exist in the mixture or not?" requires modifying the traditional SV paradigm. Specifically, utterances do not need to be mapped onto embeddings that are trained to discriminate between speakers. Instead, we propose to focus on identifying commonalities between speech from the target speaker and the mixture. Information about the target speaker can be obtained from the speaker's enrollment data.

To address speaker verification in a multi-speaker environment, we propose a binary classification deep neural network (DNN) that detects the presence of the target speaker based on a frame-level fusion of the target and the mixture features. The proposed system takes a pair of inputs: the mixture and a reference speech uttered by the target speaker. First, the reference and the mixture signals are processed using two different networks to extract their frame-level features. For the reference speech, these features are aggregated temporally

*A joint institution of the Friedrich-Alexander-University Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits (IIS).

to form an embedding that encodes the characteristics of the target. This embedding is then fused with the time-variant features of the mixture. Subsequently, the fused features are used to verify whether the target speaker is active in the mixture or not. Contrary to the previous works, the proposed approach specifically addresses the concurrent speakers problem in a direct manner without entailing speaker diarization, speaker separation, or an enhancement front-end.

The remainder of the paper is structured as follows: In Section II, we review a state-of-the-art SV approach which is later used as a baseline. In Section III, we introduce the proposed approach. In Section IV, we discuss the data generation and experimental setup. In Section V, we evaluate the proposed method and discuss the results. Finally, in Section VI, we conclude the paper.

II. Classical Speaker Verification

Existing systems address the SV problem in two stages. First, embeddings are trained to discriminate between the speech signals based on the speaker's identity. After the training is finished, a scoring backend is used to determine whether the enrollment-time embedding of the target speaker matches the test-time embedding or not. This high-level picture is depicted in Figure 1.

The SV problem formulation in the classical settings can be stated as follows. Let $f : \mathbb{R}^T \mapsto \mathbb{R}^D$ be the embedding mapping, and $s : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}$ be a similarity (scoring) function, where T is the number of time-domain samples and D is the embeddings dimensionality. The embeddings f are assumed to be learned by a neural network, and s can be a predefined or trainable function. The goal is to find f and s such that

$$\begin{aligned} s(f(x^i_{\text{enroll}}), f(x^j_{\text{test}})) &\geqslant \theta \quad \text{if } j = i, \\ s(f(x^i_{\text{enroll}}), f(x^j_{\text{test}})) &< \theta \quad \forall j \neq i, \end{aligned} \tag{1}$$

where $x_{\text{enroll}} \in \mathbb{R}^{T_{\text{enroll}}}$ is the pre-recorded enrollment utterance used to model the target speaker, $x_{\text{test}} \in \mathbb{R}^{T_{\text{test}}}$ is the test-time utterance that needs to be verified, $\theta \in \mathbb{R}$ is the decision threshold, and i, j are speakers' labels.

An issue with the aforementioned approach emerges when x_{test} is a mixture containing speech from multiple concurrent speakers. That is,

$$x_{\text{test}} = x^i + \sum_{j \in \mathcal{Q}} x^j + v, \tag{2}$$

where $\mathcal{Q}$ is a set of speaker labels for which $\mathcal{Q} \cup i = \emptyset$, and $v \in \mathbb{R}^{T_{\text{test}}}$ is an additive non-speech interference. In the following, we assume that x^i and at least one x^j with $j \neq i$ are concurrently active with comparable energy levels. In this case, reliable detection of the i-th speaker from a set of $|\mathcal{Q}|+1$ speakers from $f(x_{\text{test}})$ is difficult since $f(x_{\text{test}})$ provides an embedding for the entire mixture and not only for the i-th speaker. We hypothesize that in order to improve the target speaker detection, information about the target speaker (i.e., x^i_{enroll}) should be involved before producing the utterance-level representation, i.e., prior to the temporal aggregation step.

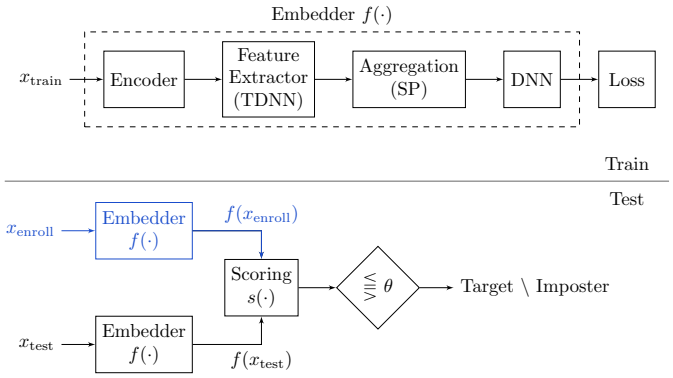

Fig. 1: SV approach based on the x-vector baseline. In the training phase, $f(\cdot)$ is trained by optimizing a loss to discriminate between speakers. After training, speakers are registered given their enrollment data x_{enroll}. At test time, $f(x_{\text{test}})$ is computed and scored via $s(\cdot)$ against $f(x_{\text{enroll}})$ of the claimed target to determine if the speaker is the same or not.

This hypothesis is the foundation of the proposed approach presented in Section III.

III. Proposed Method

A. Problem Formulation

To generalize to multi-speaker environments, we formulate the SV problem as a detection problem. Namely, let us assume that the SV system takes a pair of inputs x^i_{enroll} and x_{test} given by (2). Based on this, we define the following hypotheses:

$$\begin{aligned} &\text{Hypothesis } H_1\text{: } i\text{-th speaker is present } (x^i \neq 0) \\ &\text{Hypothesis } H_0\text{: } i\text{-th speaker is absent } (x^i = 0). \end{aligned} \tag{3}$$

The goal of the proposed system is to find a model g that estimates the target presence probability

$$g(x_{\text{test}}, x^i_{\text{enroll}}) = p(H_1 | x_{\text{test}}, x^i_{\text{enroll}}). \tag{4}$$

In this formulation, the network's output directly corresponds to the detection score. Therefore, the scoring backend s is not required.

B. System Overview

The idea behind the proposed approach is to include target-specific information in the system before temporally aggregating the frame-level features. Features of the target speaker and the mixture are extracted from x^i_{enroll} and x_{test}, respectively, fused to produce target-enhanced features, and then passed to the aggregation and other processing units, as shown in Figure 2. We believe it is essential to perform the feature fusion before temporal aggregation to emphasize the frame-level features with dominant target activity. Pooling can then be performed on the target-enhanced features to aggregate the statistics over multiple frames. The described logic applies to the features of the mixture, but not to the features of the enrollment data that contains information about the target

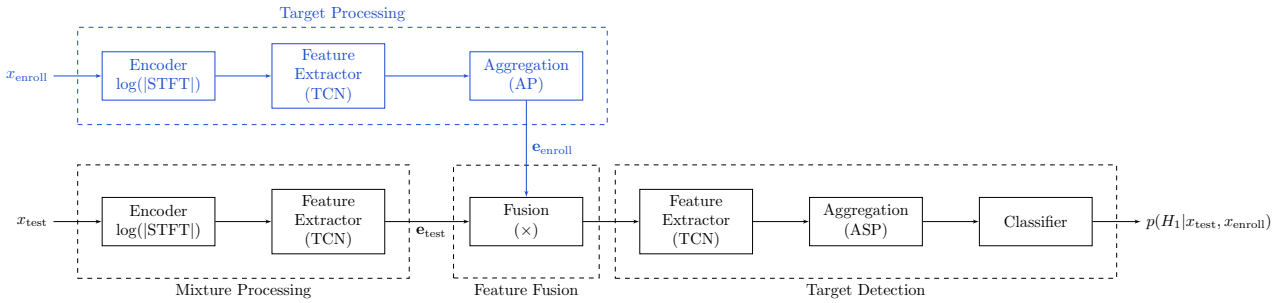

Fig. 2: Proposed temporal feature fusion SV system. For clarity, $\mathbf{e}_{\text{enroll}} \in \mathbb{R}^N$ and $\mathbf{e}_{\text{test}} \in \mathbb{R}^N \times \mathbb{R}^L$, where N is the number of short-time Fourier transform (STFT) bases and L is the number of time-frames of x_{test} in the STFT domain.

TABLE I: Architectural details of the proposed model with approx. 0.959 M parameters. The TCNs hyperparameters naming convention follow [18]. The LinearBlock is a linear layer followed by ReLU and batch normalization.

Module	Hyperparameters	Value(s)
TCNs	$\{N, B, H, P, X, R\}$	$\{257, 32, 64, 3, 6, 3\}$
ASP	Attention channels	128
Classifier	Linear	(514, 257)
	LinearBlock	(257, 257)
	LinearBlock	(257, 257)
	Linear + Sigmoid	(257, 1)

speaker only. Hence, it is more intuitive to perform feature fusion using time-varying mixture features and time-invariant target features. This design choice is also found in other systems that are used to address related problems [16, 17].

In our work, we assume that the reference information of the target contains mostly clean speech. This assumption is valid at test time as the target speech can be obtained from the speaker's enrollment data. Sections III-C and III-D describe the employed model and the proposed training and testing protocols, respectively.

C. Model

The proposed model can be conceptualized in three steps: 1) target and mixture feature extraction, 2) temporal feature fusion, and 3) target speaker detection. Figure 2 depicts the model structure and Table I details the architecture. The frame-level features from the reference and the mixture signals are extracted using temporal convolutional network (TCN) models, which have been shown to be successful feature extractors [18, 19]. The learned features are extracted from the log spectrograms of the inputs. In this step, only the frame-level features of the target are aggregated into an utterance-level representation $\mathbf{e}_{\text{enroll}}$ using average pooling (AP). After that, the extracted features are fused by element-wise multiplication. The fused features are processed frame-wise via a third TCN , aggregated using attentive statistics pooling (ASP) [9], and finally passed into a classifier to predict if the target is detected or not.

D. Training and Testing

We used the binary cross-entropy (BCE) loss to train the proposed model. Assuming that the pair label y equals 1 when H_1 is true and 0 otherwise, the training loss is defined as

$$L_{\text{BCE}} = (y-1)\log(1 - p(y|x^i_{\text{enroll}}, x_{\text{test}})) - y\log(1 - p(y|x^i_{\text{enroll}}, x_{\text{test}})). \quad (5)$$

During training, x_{enroll} and x_{test} are sampled from the training data, and not from the enrollment and test data, respectively.

The inference process is naturally very similar to the training, except that no loss needs to be computed. That is, detecting whether or not the target speaker exists within the mixture is directly determined by the network's output.

IV. Experimental Setup and Data Generation

The data in the experiments is based on VoxCeleb1 [20] and VoxCeleb2 [21]. The total number of speakers in the combined development sets is 7205 and the speech has a 16 kHz sampling rate. The development set is split into 90% training and 10% validation. Speech segments in either set have a fixed length of 3 seconds. Each sample is augmented by noise and reverberation to produce three additional samples. The first augmented sample is a reverberant version of the speech signal. The second augmented sample is generated by adding noise from MUSAN [22]. The last augmented sample is generated by adding another speaker randomly selected from the current batch. Interference in the augmented samples was artificially reverberated with a probability of 0.2. All the room impulse responses we used are described in [23]. The signal-to-interference ratio (SIR) and signal-to-noise ratio (SNR) are uniformly sampled from the interval of $[0, 15]$.

For evaluation, we use three different test sets. The first is the Voxceleb1 test set (cleaned), which we refer to as the raw set $\mathcal{R}$. The second, referred to as $\mathcal{N}$, is derived from $\mathcal{R}$ by corrupting it with additive non-speech from FSDnoisy18k test set [24]. The third, referred to as $\mathcal{I}$, is derived from $\mathcal{R}$ by corrupting it with an interfering speaker sampled randomly from the validation set. The SNR and the SIR ranges are both uniformly sampled from the interval of $[0, 5]$. The number of interfering speakers is set to 1.

TABLE II: EER[%] under various settings. Sets $\mathcal{R}$ is the Voxceleb1-test set (Cleaned) while $\mathcal{N}$ and $\mathcal{I}$ are contaminated versions as described in Section IV.

System	Test Set		
	$\mathcal{R}$	$\mathcal{N}$	$\mathcal{I}$
X-vector	9.05	11.77	19.48
X-vector (pre-trained)	8.87	11.41	20.60
Proposed	**7.81**	**8.70**	**13.58**

Our experiments were based on the PyTorch framework [25]. For the proposed model, the data was encoded using the log spectrogram with 512 discrete Fourier-transform points and a window length of 32 ms with 50% overlap. We used the Adam optimizer [26] with a learning rate $1e^{-4}$. The learning rate was reduced on a plateau by a factor 0.5 every epoch the error rate did not improve on the validation set. The number of epochs was 20 and the batch size is 42.

The x-vector system in [27] is used as a baseline. Furthermore, for a fair comparison, we trained another x-vector using the same training set and augmentations used to train the proposed model. In the trained x-vector, we maintained almost identical hyperparameters as in the pre-trained system. However, due to the hardware limitations, we reduced the batch size in the trained model from 256 to 150. The SV trials of the x-vector were scored using cosine similarity.

V. Performance Evaluation

We report the results using the equal error rate (EER). Both the proposed model and the baseline were tested on the three sets $\mathcal{R}$, $\mathcal{N}$, and $\mathcal{I}$. Table II outlines the results on the three test sets.

The proposed model outperforms the x-vector models by a small margin on the official splits in $\mathcal{R}$. Notable gains are achieved over $\mathcal{N}$ and $\mathcal{I}$ compared to the baseline. Both the proposed system and baselines demonstrate a reasonable robustness to noise, with a drop of 11.4% and 32.6% in their respective EERs relative to $\mathcal{R}$. In the case of $\mathcal{I}$, these figures further drop to 73.9% and 114.7%, respectively, increasing the performance gap between the proposed and the x-vector models. This gap is visualized in the detection error tradeoff (DET) curves shown in Figures (3a) and (3b).

While poor x-vector performance on $\mathcal{I}$ is expected, it is surprising that the trained and pre-trained models behave similarly. For the latter, it is conceivable that the model did not generalize for $\mathcal{I}$ since the condition was unseen in training. This, however, is not the case for the trained x-vector; every training batch was augmented by samples containing interfering speakers. Yet, the x-vector's capacity to accommodate interfering speakers did not improve albeit explicitly trained to fit this scenario. This finding supports our hypothesis: learning robust speaker embeddings is challenging in a multi-speaker environment.

For the other cases, namely $\mathcal{R}$ and $\mathcal{N}$, the performance similarity is expected. Both models (trained and pre-trained x-vector) reasonably generalize to noisy environments as noise augmentation is part of the training of both models. The slight difference in the EERs can be understood due to the difference in the batch size, data preparation, and the seed.

(a) Official VoxCeleb1 splits $\mathcal{R}$ (b) Multi-speaker scenario $\mathcal{I}$

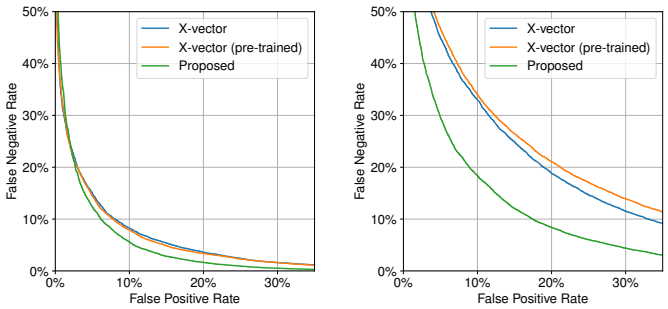

Fig. 3: DET curves computed for the test sets $\mathcal{R}$ and $\mathcal{I}$. A performance gap is observed between the proposed system and the x-vector in the presence of interfering speakers (bottom). The curves computed for $\mathcal{N}$ (omitted due to the space constraint) lies somewhere in-between $\mathcal{R}$ and $\mathcal{I}$.

VI. Conclusion

We introduced a speaker verification system a multi-speaker environment using temporal feature fusion and a reference of the target speaker. The fusion of the target information with the frame-level features of the mixture allows detecting whether the target exists in the mixture or not. We have shown that the proposed system is generally more robust against interfering signals than the x-vector baseline. The performance of the x-vector was shown to suffer in the presence of concurrent speakers, even when explicitly trained to generalize to this scenario. We have demonstrated that the proposed speaker verification system achieves a 30.3% relative improvement over the x-vector in multi-speaker environments and 13.7% on the official VoxCeleb1 splits.

References

[1] C. Zhang, K. Koishida, and J. H. L. Hansen, "Text-Independent Speaker Verification Based on Triplet Convolutional Neural Network Embeddings," *IEEE/ACM Trans. Netw.*, vol. 26, no. 9, pp. 1633–1644, Sep. 2018.

[2] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[3] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 999–1003.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *Proc. IEEE Intl. Conf.*

on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, Apr. 2018, pp. 5329–5333.

[5] W. Lin, M.-W. Mak, and J.-T. Chien, "Strategies for End-to-End Text-Independent Speaker Verification," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 4308–4312.

[6] M.-W. Mak and J.-T. Chien, *Robust Speaker Verification*. Cambridge University Press, 2020, pp. 169—-216.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, Jul. 2016, pp. 770–778.

[8] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3830–3834.

[9] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech*, 2018, pp. 3573–3577.

[10] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. P. García-Perera, D. Povey, P. A. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-Art Speaker Recognition for Telephone and Video Speech: The JHU-MIT Submission for NIST SRE18," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 1488–1492.

[11] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep Speaker Embedding Extraction with Channel-Wise Feature Responses and Additive Supervision Softmax Loss Function," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 2883–2887.

[12] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-speaker Conversations Using X-vectors," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, Apr. 2019, pp. 5796–5800.

[13] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[14] K. A. Lee, H. Yamamoto, K. Okabe, Q. Wang, L. Guo, T. Koshinaka, J. Zhang, and K. Shinoda, "The NEC-TT 2018 Speaker Verification System," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 4355–4359.

[15] A. F. Martin and M. A. Przybocki, "Speaker Recognition in a Multi-Speaker Environment," in *Proc. Interspeech*, Aalborg, Denmark, Sep. 2001, pp. 787–790.

[16] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 2728–2732.

[17] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, "SpeakerBeam: Speaker Aware Neural Network for Target Speaker Extraction in Speech Mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, Aug. 2019.

[18] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Trans. Netw.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[19] S. Han, J. Byun, and J. W. Shin, "Time-Domain Speaker Verification Using Temporal Convolutional Networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, Jun. 2021, pp. 6688–6692.

[20] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 2616–2620.

[21] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech*, Hyderabad, India, Sep. 2018, pp. 1086–1090.

[22] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *CoRR*, 2015.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 5220–5224.

[24] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Learning Sound Event Classifiers from Web Audio with Noisy Labels," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 21–25.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. Neural Information Processing Conf.* New York, NY, USA: Curran Associates, Inc., Dec. 2019.

[26] J. Kingma, "Adam: A Method for Stochastic Optimization," in *Intl. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[27] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A General-Purpose Speech Toolkit," 2021.