Interspeech 2024
1-5 September 2024, Kos, Greece

ESPnet-SPK: full pipeline speaker embedding toolkit with reproducible recipes, self-supervised front-ends, and off-the-shelf models

Jee-weon Jung[1,*], *Wangyou Zhang*[1,*], *Jiatong Shi*[1,*], *Zakaria Aldeneh*[2], *Takuya Higuchi*[2], *Alex Gichamba*[1], *Barry-John Theobald*[2], *Ahmed Hussen Abdelaziz*[2], *Shinji Watanabe*[1]

[1]Carnegie Mellon University, USA [2]Apple, USA

jeeweonj@ieee.org, shinjiw@ieee.org

Abstract

This paper introduces ESPnet-SPK, a toolkit designed for training and utilizing speaker embedding extractors. It offers an open-source platform, facilitating effortless construction of models ranging from the x-vector to the SKA-TDNN, thanks to its modular architecture that simplifies the development of variants. The toolkit advances the use of speaker embeddings across various tasks where outdated embeddings are often employed, enabling the broader research community to use advanced speaker embeddings effortlessly. Pre-trained extractors are readily available for off-the-shelf use. The toolkit also supports integration with various self-supervised learning features. ESPnet-SPK features over 30 recipes: seven speaker verification recipes, including reproducible WavLM-ECAPA with an EER of 0.39% on the Vox1-O benchmark and diverse downstream tasks, including text-to-speech and target speaker extraction. It even supports speaker similarity evaluation for singing voice synthesis and more.

Index Terms: speaker verification, speaker recognition, toolkit

1. Introduction

In recent years, the rapid advancements in speech technologies, including speaker recognition with deep learning, have yielded robust speaker embeddings [1–3].These compact representations of unique speaker characteristics are pivotal in diverse tasks, including speaker verification, diarization [4, 5], Text-To-Speech (TTS) [6], and other speech-related tasks [7–10]. With growing demand for speaker embedding extractors, numerous open-source toolkits have emerged to streamline the training and deployment of these models [11–18].

Table 1 shows prominent speaker recognition toolkits. Though these toolkits have substantially contributed to the community, they often fail to address its rapidly evolving requirements. Many lack a comprehensive set-up that integrates data preprocessing, feature extraction, and training/inference design, limiting researchers' access to state-of-the-art (SOTA) models. Consequently, inferior speaker embeddings are frequently used in downstream tasks [6, 19]. Furthermore, despite the proven effectiveness of the features of Self-Supervised Learning (SSL) models [20, 21], recipes for incorporating these features into speaker embedding extractor development are notably absent.

To complement these existing toolkits, we introduce ESPnet-SPK, a new open-source toolkit for speaker embedding extraction designed to meet the research community's evolving needs. ESPnet-SPK aims to address multiple objectives: foremost, it offers a unified platform for speaker recognition researchers to develop, compare, and evaluate models across diverse datasets in a unified environment. It features a wide array of models, from the x-vector [1] to the advanced SKA-TDNN [22]. Its modular composition facilitates novel model development; the combination of modules can result in hundreds of different models. Additionally, it supports evaluations for the newly emerging Spoofing-robust Automatic Speaker Verification (SASV) task [23], highlighting the toolkit's scalability and adaptability.

Table 1: *Open-source speaker verification toolkits. △: pre-trained models exist, but not prepared for off-the-shelf usage. †: currently does not support a SSL feature-based speaker verification recipe, but possible with modifications.*

Name	Datasets	SSL front-end	Off-the-shelf
ASV-subtools [14]	Multiple	✗	✗
NeMo [17]	VoxCeleb	✗	✗
Kaldi [11]	Multiple	✗	△
VoxCeleb_Trainer [12]	VoxCeleb	✗	△
Wespeaker [13]	Multiple	✗	✓
3D-Speaker [18]	Multiple	✗	△
PaddleSpeech [15]	VoxCeleb	✓†	✓
SpeechBrain [16]	VoxCeleb	✓†	✓
ESPnet-SPK	Multiple	✓	✓

Moreover, the toolkit grants the broader speech community easy access to the latest models for their research, providing pre-trained speaker embedding extractors and the necessary tools for effortless "off-the-shelf" applications. Speaker embeddings generated by ESPnet-SPK are versatile and *can be applied to a wide range of speech-processing tasks beyond speaker verification*; it currently supports over 30 recipes on diverse downstream tasks, including TTS, TSE, target speech recognition, and more. We demonstrate its utility in two specific tasks with reproducible ESPnet recipes: multi-speaker TTS and Target Speaker Extraction (TSE).

Additionally, ESPnet-SPK incorporates SSL features into speaker embedding development, enhancing task performance. This integration augments the *number of available model architectures further to thousands*, utilizing S3PRL[1]. We showcase a recipe with an Equal Error Rate (EER) of 0.39% on the Vox1-O evaluation protocol, utilizing WavLM-Large [21] and ECAPA-TDNN [24]. This capability distinguishes ESPnet-SPK from other toolkits, like PaddleSpeech and SpeechBrain [16], offering broader and more competitive model support beyond a single recipe and exemplifying its potential to meet and exceed the evolving benchmarks of the research community.

2. Framework

2.1. Design

The ESPnet-SPK toolkit consists of two main components: the speaker verification recipe and off-the-shelf speaker embed-

[1]https://github.com/s3prl/s3prl.

 10.21437/Interspeech.2024-1345

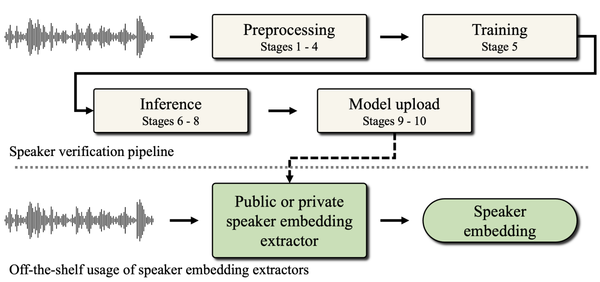

Figure 1: *Process Pipeline of ESPnet-SPK, structured in multiple stages akin to the Kaldi [11] speech processing toolkit. The top section outlines stages 1 through 10, the speaker verification process, with stages 9 and 10 dedicated to optionally publishing trained speaker embedding extractors. Furthermore, it highlights the ease of using publicly available embedding extractors in an off-the-shelf manner.*

ding extraction. The speaker verification recipe comprises eight stages targeted to help speaker recognition researchers and detailed in Section 2.2. The off-the-shelf component, outlined in Section 2.3, addresses the deployment of trained speaker embedding extractors, making them readily available for public use. This sub-section is specifically designed to support researchers outside the speaker recognition field on top of those within it, illustrating the framework's comprehensive approach. Figure 1 provides an overview of the entire toolkit.

2.2. Speaker verification recipe

Preprocessing. The preprocessing phase is divided into four stages, each with a distinct function:

- Stage 1: Downloading and preparing datasets, including those for augmentation;
- Stage 2: Applying speed perturbation to audio files [25];
- Stage 3: Formatting the data for processing;
- Stage 4: Calculating relevant statistics.

These initial stages are consistent with the broader ESPnet framework, and we direct the readers to Watanabe *et al.* [26] for comprehensive details. A notable distinction in our approach, as inspired by Yamamoto *et al.* [27] and Chen *et al.* [28], is treating speakers in speed-perturbed utterances as distinct entities rather than identical ones. This augmentation method has proven beneficial for enhancing speaker verification performance.

Training. Stage 5 executes the core training process. Through a centralized `yaml`-style configuration file, we specify the model architecture, objective function, optimizer, and additional training hyperparameters. The configuration file is then used to train the speaker embedding extractor efficiently.

Supported models. ESPnet-SPK currently supports five predefined model architectures: x-vector [1], MFA-Conformer [29], ECAPA-TDNN [24], RawNet3 [30], and SKA-TDNN [22]. The number of architectures, nevertheless, can increase to thousands with different module combinations, SSL front-ends, and encoders from the other ESPnet ecosystem. While we omit detailed architectural descriptions due to space constraints, we direct readers to the respective publications for in-depth information.

Modularized sub-components. ESPnet-SPK organizes the speaker embedding extractor into four key sub-components as illustrated in Figure 2: front-end, encoder, pooling, and projector. The *front-end* processes raw waveforms to generate a

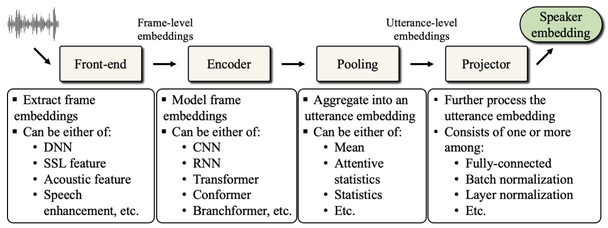

Figure 2: *Illustration of the modular sub-components of the speaker embedding extractor. Users can effortlessly construct thousands of model architectures in the configuration file by combining these sub-components.*

```
import numpy as np
from espnet2.bin.spk_inference import
     Speech2Embedding

# from uploaded models
speech2spk_embed = Speech2Embedding.from_pretrained
     (model_tag="espnet/voxcelebs12_rawnet3")
embedding = speech2spk_embed(np.zeros(16500))

# from checkpoints trained by oneself
speech2spk_embed = Speech2Embedding(model_file="
     model.pth", train_config="config.yaml")
embedding = speech2spk_embed(np.zeros(32000))
```

Figure 3: *Sample Code demonstrating the use of public and custom speaker embedding extractors of ESPnet-SPK.*

sequence of input features. For models like RawNet3 that directly process raw waveforms, the front-end includes layers such as Sinc-conv with normalization to provide an integrated format for the encoder input. The *encoder* is the principal block for frame-level feature modeling, e.g., convolutional or Conformer blocks. The *pooling* contains aggregation methods, including the widely used attentive statistics pooling [31]. The *projector* consists of additional layers, typically including fully-connected, batch normalization [32], or layer normalization layers [33].

Objective functions, optimizers, and learning rate schedulers. ESPnet-SPK accommodates a wide array of objective functions, from traditional cross-entropy to advanced AAM-Softmax [34], enhanced with sub-center and inter top-k techniques [35]. It shares the same optimizers and learning rate schedulers as the broader ESPnet ecosystem, ensuring access to various established configurations.

Inference. The inference phase is segmented into stages 6, 7, and 8, each serving a specific purpose:

- Stage 6: Extracting speaker embeddings from all utterances within the test set;
- Stage 7: Applying post-normalization and enhancement processes at the score level;
- Stage 8: Calculating the metrics.

Stage 6 additionally extracts embeddings for utterances used in subsequent processing in Stage 7. This stage is crucial for preparing cohort sets for score normalization and quality measurement. **Stage 7** applies additional processing in the score level. Once all trial scores are derived, configurations determine whether to apply score normalization and quality measure functions. For score normalization, we support AS-norm [36]; for quality measure function, we implement those mentioned in [37]. **Stage 8** lastly calculates the metrics, EER, and minimum Detection Cost Function (minDCF). For the minDCF, we use a p_{trg}, $C_{\text{false\_alarm}}$, and C_{miss} of 0.05, 1, and 1 respectively.

2.3. Off-the-shelf usage

After completing the training and inference phases, models can be shared publicly through stages 9 and 10:

- Stage 9: Prepares the model for distribution;
- Stage 10: Uploads the model to Huggingface.

For the models already made publicly available, anyone can use them off the shelf. Users can easily upload their own models by simply specifying the model name.[2] This functionality allows individuals within and outside the speaker recognition community to utilize or contribute to the ESPnet-SPK ecosystem easily. Figure 3 demonstrates extracting a speaker embedding from a pre-trained RawNet3 model in under ten lines of Python code, showcasing the framework's user-friendliness.

3. Experiments

ESPnet-SPK has been developed using PyTorch and lives at `https://github.com/espnet/espnet`.

3.1. Corpus

This paper investigates the widely used VoxCeleb 1&2 [38, 39], the newly introduced VoxBlink [40], and the ASVspoof 2019 [41] corpora. Although the toolkit features several additional recipes, including VOiCES and the SdSV Challenge, this paper does not cover them due to space constraints.

VoxCeleb 1&2 includes $7,333$ speakers with over 1.2M utterances. Our experiments primarily utilize the combined development sets, comprising approximately $2,690$ hours of speech from $7,205$ speakers. The Vox1-O evaluation protocol, involving around 37k trials from 40 speakers in the VoxCeleb1 test set, serves as our primary benchmark.

VoxBlink comprises two subsets: the full set, with 1.45 million utterances from 38k speakers, and the clean set, a subset with 1.02 million utterances from 18k speakers. VoxBlink complements VoxCeleb by focusing on a broader demographic from YouTube videos, offering a contrast to VoxCeleb's celebrity focus.

ASVspoof 2019 logical access test set is used to show the Spoofing-robust Automatic Speaker Verification (SASV) [23] capabilities of the embedding extractors, highlighting the toolkit's adaptability to recent trends in speaker verification.

3.2. Vox1-O and ASVspoof 2019 benchmarks

Table 2 displays the benchmark results across various representative models using the Vox1-O and SASV 2022 evaluation protocols. Input features were primarily mel-spectrograms derived from 3 seconds of speech, except for RawNet3, which processes raw waveforms directly. We adopted the AAM-softmax with sub-center, and inter top-k penalty loss functions [35] with an Adam optimizer.

In the Vox1-O benchmark, SKA-TDNN and RawNet3 emerged as top performers among the five models presented, all accessible off-the-shelf. The SASV 2022 benchmark revealed the vulnerability of contemporary speaker verification systems to spoofing attacks, underscoring the critical need for future research in SASV. RawNet3 demonstrated the highest resistance to spoofing among the evaluated models, which is consistent with studies in audio anti-spoofing, where raw waveform-based models demonstrate superior performance [42].

[2]Available list of models live at `https://github.com/espnet/espnet/egs2/[corpus]/spk1/README.md`.

Table 2: *Comparison of models on the Vox1-O and SASV 2022 evaluation protocols. Lower is better for all three metrics.*

Model	EER (%) ↓	minDCF ↓	SASV-EER (%) ↓
x-vector [1]	1.81	0.1251	25.84
MFA-Conformer [29]	0.86	0.0627	24.71
ECAPA-TDNN [24]	0.85	0.0666	26.12
RawNet3 [30]	0.73	0.0581	**17.41**
SKA-TDNN [22]	**0.72**	**0.0457**	21.75

Table 3: *Comparison of ECAPA- and SKA-TDNN performance using different front-ends on the Vox1-O evaluation protocol. The table compares mel-spectrogram, WavLM (fixed), and WavLM (jointly fine-tuned). Reported in EERs (%). "Mel-spec": mel-spectrogram, "tuned": jointly fine-tuned.*

Model	Front-end		
	Mel-spec	WavLM (fixed)	WavLM (tuned)
ECAPA-TDNN	0.85	0.60	**0.39**
SKA-TDNN	0.72	0.56	0.51

3.3. SSL front-end replacements

Table 3 presents the impact of substituting the front-end from mel-spectrogram to WavLM-Large SSL model. The integration with S3PRL enables straightforward access to over 100 SSL models through minor adjustments in the `yaml` configuration file. WavLM was chosen for its exceptional performance in speaker-related tasks, as demonstrated in the SUPERB benchmark [20]. Our experiments used ECAPA-TDNN and SKA-TDNN in three configurations: traditional mel-spectrogram front-end, fixed WavLM front-end, and jointly fine-tuned WavLM front-end, keeping all other hyperparameters consistent with Section 3.2.

The results indicate a consistent trend: WavLM surpasses the mel-spectrogram, and joint fine-tuning of the SSL front-end yields better results than the fixed approach. Specifically, for ECAPA-TDNN, ESPnet-SPK achieved an EER of 0.39%, closely mirroring the 0.38% EER reported in the original work [21], marking the first reproducible benchmark of this performance. However, for SKA-TDNN, the benefits of jointly fine-tuning WavLM were more modest. This could be due to SKA-TDNN initially processing inputs through the "fcwSKA Block," which could align more effectively with traditional acoustic features with linearly positioned frequency bands.

3.4. Recipes on VoxBlink

Table 4 presents the experimental results on the VoxBlink corpus [40]. While the authors released the full- and the clean-sets, this paper marks the first instance of model performance being reported on the full set, highlighting ESPnet-SPK's role as a pioneer in providing open-source recipes. Our investigation included training with separate sets (rows 2 and 3) and their integration with the VoxCeleb1&2 development sets (rows 4 and 5). The data quality proved pivotal; the model trained on the clean set outperformed the full set despite the latter's larger volume of data (2.1k vs 1.6k hours). Contrary to the findings reported in the original VoxBlink study, where combining VoxCeleb to the training data yielded a 10% improvement, our results showed superior performance when we trained the model with only VoxCeleb data. This discrepancy could stem from our inclusion of the VoxCeleb1 development set.

3.5. Downstream tasks demonstration

TTS results. Adopting different speaker embeddings affects the performance of multi-speaker TTS systems [6, 19]. Never-

Table 4: *Results from experiments using the VoxBlink dataset [40]. This is the inaugural report of model training exclusively with the VoxBlink-full set. The model architecture is fixed to RawNet3. "Vb-full": VoxBlink-full, "Vb-cln": VoxBlink-clean.*

Voxcelebs	Vb-full	Vb-cln	EER (%) ↓	minDCF ↓
✓	-	-	**0.73**	0.0581
-	✓	-	2.68	0.1893
-	-	✓	2.51	0.1858
✓	✓	-	0.78	0.0655
✓	-	✓	0.77	**0.0556**

Table 5: *Results from TTS experiments using the VITS model [43] on the VCTK corpus [44], employing a range of speaker embeddings from ESPnet-SPK. Reference scores for Mean Opinion Score (MOS) and Speaker similarity MOS (SMOS) are* 4.04 *and* 4.84*, respectively.*

Embedding	MCD ↓	WER (%) ↓	MOS ↑	SMOS ↑
x-vector [1, 16]	6.76	9.83	3.39	**3.92**
Rawnet3 [30]	6.76	**7.67**	**3.49**	3.84
WavLM(fixed)+ECAPA	**6.69**	9.30	3.42	3.82
WavLM(tuned)+ECAPA	6.75	8.04	3.45	3.85

theless, the specific impact of different pre-trained speaker embedding extractors on TTS still needs to be explored. We evaluate the performance of four pre-trained speaker embeddings by integrating them with a VITS model [43] on the VCTK corpus [44]. Notably, the ESPnet ecosystem provides more than 20 TTS recipes, which can leverage any speaker embedding extractor of ESPnet-SPK, including LibriTTS and AISHELL-3.

Our evaluation uses both objective and subjective metrics, including Mel Cepstral Distortion (MCD), Word Error Rate (WER) assessed by Whisper-small [45], UTMOS [46], Mean Opinion Score (MOS), and Speaker Similarity MOS (SMOS). Each utterance received six votes from English speakers for both MOS and SMOS evaluations. Results, presented in Table 5, illustrate the ease of comparing different speaker embeddings through ESPnet-SPK. RawNet3 performed best in MOS and WER evaluations, while the top systems for other metrics varied.

TSE results. Speaker embeddings play a crucial role in TSE, where they act as a reference to isolate the target speaker's voice from a mix of multiple speakers. The selection of an optimal speaker embedding is essential for effective TSE. We evaluate the Perceptual Evaluation of Speech Quality (PESQ) and Signal-to-Distortion Ratio (SDR) performance of four distinct speaker embeddings in TSE using the WSJ0-2mix benchmark [47].[3] We use the popular `min` version of WSJ0-2mix, where we ensure that each sample contains only fully overlapping speech samples with careful trimming. We exploit the ESPnet-SE recipe, demonstrating the ease of connectivity within the ESPnet ecosystem. Our experiments with the TD-SpeakerBeam model [48] involve fusing the speaker embedding after the initial temporal convolutional network block. Our primary aim is not to achieve peak performance but to demonstrate the straightforward integration of developed speaker embeddings into various related tasks.

The findings, detailed in Table 6, demonstrate the facility of ESPnet-SPK for comparing various speaker embeddings through ESPnet-SPK in TSE studies. Despite its relatively lower performance in speaker verification, we discovered that the x-vector (512-dimensional) shows the highest TSE performance across both validation and test sets. This observation aligns with insights from [49, 50]. In contrast, speaker embeddings with 192 dimensions yield inferior results on the test dataset, suggesting vulnerability to target speaker confusion issues. We analyze that the x-vector's higher dimensionality may encapsulate a broader range of information, potentially harmful to speaker verification yet beneficial for downstream tasks.

[3]We also provide a recipe on LibriMix, another popular TSE corpus.

Table 6: *TSE results on WSJ0-2mix (*`min` *mode) using the TD-SpeakerBeam model [48] with various speaker embeddings.*

Embedding extractor	PESQ ↑		SDR (dB) ↑	
	validation	test	validation	test
x-vector [1, 16]	**3.57**	**3.25**	**18.96**	**15.70**
RawNet3 [30]	3.54	2.64	18.65	9.21
WavLM-Large(fixed)+ECAPA	3.53	2.77	18.60	10.60
WavLM-Large(tuned)+ECAPA	3.53	2.79	18.68	11.27

Table 7: *Ablations on selected components. Objective function and train data is ablated on RawNet3. Model architecture is ablated on ECAPA-TDNN.*

Model	EER (%) ↓	minDCF ↓
RawNet3	0.73	0.0581
→ w/o sub-center & inter top-k	0.79	0.0626
→ w/o VoxCeleb1 dev	0.85	0.0698
WavLM-Large (fixed) + ECAPA	0.60	0.0446
→ w/o VoxCeleb2 dev	1.33	0.1157
→ w/ identity encoder	1.88	0.1363

3.6. Additional results on loss functions, training data, and architecture

Lastly, Table 7 presents a series of ablations assessing the impact of varying loss functions, training datasets, and architectures. First, we examine two modifications to RawNet3: altering the objective function and adjusting the training dataset. Substituting the complex sub-center AAM-softmax and inter top-k with only AAM-softmax increased the EER to 0.79%. Exclusively using VoxCeleb2-dev for training by omitting VoxCeleb1-dev further raised the EER to 0.85%. These findings underscore the significance of objective functions and training data for optimal performance. Again, the experiments demonstrate the ease of using different loss functions, training data, and architectures within the toolkit.

Subsequent ablations focused on models featuring SSL front-end features inspired by [51]. Training exclusively with VoxCeleb1-dev yielded an EER of 1.33%. Removing the ECAPA-TDNN encoder to streamline the architecture to WavLM, attentive statistics pooling, and a projection layer led to an EER of 1.88%. These findings affirm the potential of SSL front-ends paired with simpler architectures and datasets for competitive outcomes, and also illustrate ESPnet-SPK's capability to facilitate exploration across various SSL model implementations.

4. Conclusion

We introduced ESPnet-SPK, a versatile open-source toolkit designed for speaker verification and embedding extraction. This toolkit provides a fully integrated pipeline featuring various reproducible recipes. It currently supports five of the most popular speaker embedding extractors and facilitates the creation of countless new model variants via its modular architecture and integration with SSL technologies. Furthermore, ESPnet-SPK offers a seamless off-the-shelf user experience, enabling a wide range of interdisciplinary applications to easily access and deploy SOTA speaker embeddings for various downstream tasks.

5. Acknowledgment

We use PSC Bridges2 and NCSA Delta via ACCESS CIS210014, by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. This work was supported by a grant from Apple, Inc. Any views, opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and should not be interpreted as reflecting the views, policies or position, either expressed or implied, of Apple Inc.

6. References

[1] D. Snyder, D. Garcia-Romero et al., "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018.

[2] J.-w. Jung et al., "RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in *Proc. Interspeech*, 2019.

[3] L. Wan, Q. Wang, A. Papir and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. ICASSP*, 2018.

[4] T. J. Park et al., "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, 2022.

[5] J.-w. Jung, H.-S. Heo et al., "In search of strong embedding extractors for speaker diarisation," in *Proc. ICASSP*. IEEE, 2023.

[6] E. Casanova et al., "YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone," in *Proc. ICML*, 2022.

[7] Q. Wang et al., "VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Proc. Interspeech*, 2019.

[8] C. Xu, W. Rao et al., "SpEx: Multi-scale time domain speaker extraction network," *IEEE/ACM Trans. ASLP.*, vol. 28, 2020.

[9] T.-h. Huang, J.-h. Lin and H.-y. Lee, "How far are we from robust voice conversion: A survey," in *Proc. SLT*, 2021.

[10] W.-C. Huang, L. P. Violeta, S. Liu et al., "The singing voice conversion challenge 2023," *arXiv:2306.14422*, 2023.

[11] D. Povey, A. Ghoshal, G. Boulianne et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[12] J. S. Chung, J. Huh, S. Mun et al., "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020.

[13] H. Wang et al., "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *Proc. ICASSP*, 2023.

[14] F. Tong, M. Zhao, J. Zhou et al., "ASV-Subtools: Open source toolkit for automatic speaker verification," in *Proc. ICASSP*, 2021.

[15] H. Zhang, T. Yuan, J. Chen et al., "PaddleSpeech: An easy-to-use all-in-one speech toolkit," in *Proc. NAACL*, 2022.

[16] M. Ravanelli, T. Parcollet, P. Plantinga et al., "SpeechBrain: A general-purpose speech toolkit," *arXiv:2106.04624*, 2021.

[17] O. Kuchaiev, J. Li, H. Nguyen et al., "NeMo: a toolkit for building ai applications using neural modules," *arXiv:1909.09577*, 2019.

[18] S. Zheng, L. Cheng et al., "3D-Speaker: A large-scale multi-device, multi-distance, and multi-dialect corpus for speech representation disentanglement," *arXiv:2306.15354*, 2023.

[19] T. Hayashi, R. Yamamoto, T. Yoshimura et al., "ESPnet2-TTS: Extending the edge of TTS research," *arXiv:2110.07840*, 2021.

[20] S. wen Yang et al., "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021.

[21] S. Chen, C. Wang et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, 2022.

[22] S. H. Mun, J.-w. Jung et al., "Frequency and multi-scale selective kernel attention for speaker verification," in *Proc. SLT*, 2022.

[23] J.-w. Jung et al., "SASV 2022: The first spoofing-aware speaker verification challenge," in *Proc. Interspeech*, 2022.

[24] B. Desplanques, J. Thienpondt et al., "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020.

[25] T. Ko, V. Peddinti, D. Povey and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015.

[26] S. Watanabe, T. Hori, S. Karita et al., "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[27] H. Yamamoto et al., "Speaker augmentation and bandwidth extension for deep speaker embedding.," in *Proc. Interspeech*, 2019.

[28] Z. Chen et al., "Build a SRE challenge system: Lessons from VoxSRC 2022 and CNSRC 2022," in *Proc. Interspeech*, 2023.

[29] Y. Zhang, Z. Lv, H. Wu et al., "MFA-Conformer: Multi-scale feature aggregation conformer for automatic speaker verification," in *Proc. Interspeech*, 2022.

[30] J.-w. Jung, Y. Kim, H.-S. Heo et al., "Pushing the limits of raw waveform speaker recognition," in *Proc. Interspeech*, 2022.

[31] K. Okabe, T. Koshinaka and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[33] J. L. Ba, J. R. Kiros and G. E. Hinton, "Layer normalization," in *NeuralIPS - Deep Learning Symposium*, 2016.

[34] J. Deng et al., "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019.

[35] M. Zhao et al., "The SpeakIn system for VoxCeleb speaker recognition challange 2021," *arXiv:2109.01989*, 2021.

[36] P. Matejka et al., "Analysis of score normalization in multilingual speaker recognition.," in *Proc. Interspeech*, 2017.

[37] N. Torgashov et al., "The ID R&D VoxCeleb speaker recognition challenge 2023 system description," *arXiv:2308.08294*, 2023.

[38] A. Nagrani, J. S. Chung and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017.

[39] J. S. Chung, A. Nagrani and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018.

[40] Y. Lin, X. Qin, G. Zhao et al., "VoxBlink: A large scale speaker verification dataset on camera," in *Proc. ICASSP*, 2024.

[41] X. Wang, J. Yamagishi, M. Todisco et al., "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, 2020.

[42] J.-w. Jung et al., "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. ICASSP*, 2022.

[43] J. Kim et al., "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021.

[44] C. Veaux, J. Yamagishi and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," CSTR, University of Edinburgh, 2019.

[45] A. Radford, J. W. Kim, T. Xu et al., "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*. PMLR, 2023.

[46] T. Saeki, D. Xin et al., "UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022," in *Proc. Interspeech*, 2022.

[47] J. R. Hershey et al., "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016.

[48] M. Delcroix, T. Ochiai, K. Zmolikova et al., "Improving speaker discrimination of target speech extraction with time-domain SpeakerBeam," in *Proc. ICASSP*, 2020.

[49] Z. Zhao et al., "Probing deep speaker embeddings for speaker-related tasks," *arXiv preprint arXiv:2212.07068*, 2022.

[50] X. Liu, X. Li and J. Serrà, "Quantitative evidence on overlooked aspects of enrollment speaker embeddings for target speaker separation," in *Proc. ICASSP*, 2023.

[51] J. Peng, T. Stafylakis, R. Gu et al., "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," in *Proc. ICASSP*, 2023.