The Biometric Vox System Description for VoxCeleb Speaker Recognition Challange 2023

Teresa Grau, Jose María Hernández

R&D department - Biometric Vox, Spain

teresa.grau@biometricvox.com, josem.hernandez@biometricvox.com

Abstract

This report describes Biometric Vox research team's submission for Tracks 1, 2 and 4 for the VoxCeleb Speaker Recognition Challange 2023 (VoxSRC2023). Our best system achieves minDCF 0.2437 and EER 4.394 in track 1, minDCF 0.2337 and EER 4.262 in track 2, DER 7.11 and JER 39.85 in track 4.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

The VoxSRC23 challenge[1] features two fully supervised and one semi-supervised domain adaptation speaker verification tracks (Track 1, Track 2 and Track 3 respectively) and one diarization track (Track 4). Organizers have set as goal for this challenge to probe how well current methods can recognise speakers from speech obtained 'in the wild'.

Our team submitted systems for Track 1, 2 and 4:

Track 1&2: Referring to how the data augmentation was applied and the features extracted, one online ResNet-based model and one offline TDNN-based model for each track were trained from scratch. Two different scoring methods were used. All the models in track 1 and 2 were trained and calibrated individually (procedure description in Section 3) and then fused using weighted linear combinations including different quality measurements (described in Section 3.3).

Track 4: A fusion of different systems submitted in previous editions of this and others diarization challenges was tested.

2. Data Resources Track 1&2

VoxCeleb2-dev [1] dataset is used to train the models following the Challenge rules for the close condition (Track 1). For the open condition (Track 2) we used: Voxceleb2-dev and a small Self-Voxceleb datasets. For validation, VoxCeleb1-test [2] set and VoxSRC23 validation sets were used.

2.1. Small BVSelf-VoxCeleb

Inspired by the idea of the VoxCeleb2 dataset collection and the winner of the past edition[3], we used a similar procedure to obtain a small dataset to increase the training data for Track 2, to which we refer as BVSelf-VoxCeleb. The small dataset contains 1000 new speakers and all the videos used are licensed under the CC BY 4.0.

We selected 1500 videos from the TEDx TALKs[2] YouTube channel, which contains mostly videos with one speaker and offers videos in several languages and with different subjects. From each video, 3-4 minutes of speech was extracted and filtered to avoid duplicated speakers and noisy videos. To filter, speaker embeddings were extracted in a 3 s window, 1.5 s hop using a TDNN as described in section 4. The embeddings were clustered using Agglomerative Hierarchical Clustering (AHC) with single linkage and the similarities between clusters were checked in order to drop duplicated speakers. Finally, segments were sampled from the filtered clusters and speakers labeled based on the video id.

2.2. Data augmentation

For data augmentation, as mention before, two methods were applied: online (sample generated during the training stage) and offline (sample generated prior to the training stage). Six augmentation strategies were adopted to train our models:

- **Music**: A single music file is randomly selected from MUSAN [4] corpus and added to the original signal (5-15dB SNR). The duration of additive noise is matched to the duration of the original signal.
- **Noise**: Randomly selected noise from MUSAN corpus added to the original recording (0-15dB SNR).
- **Speech**: Three to seven speakers are randomly picked from VoxCeleb2-dev dataset, summed together and then added to the original signal (13-20dB SNR), except for the online Track 2 model, which uses spech from MUSAN corpus.
- **Reverb**: Artificial reverberation noise created convolving the original audio with room impulse (RIRs [5]).
- **Speed**: We applied a speed augmentation[6] (0.9 and 1.1 to the original speed) increasing the number of speakers in training data by a factor of 3.
- **SpecAugment**: The only strategy that is applied online even in the offline models. We masked from 0 to 5 frames in the temporal axis and from 0 to 10 frames in the frequency axis to 40% of the batch using the SpecAug [7].

3. System Description Tracks 1&2

Referring to how the data augmentation was applied and the features extracted we trained four different systems: one online and one offline for each track.

3.1. Online Systems

3.1.1. Architecture and Training

For the online systems, we trained two resnet34 following the wespeaker [8] recipe. The acoustic features used were 80-dimensional log Mel-filterbank energies with a frame length of

[1] http://mm.kaist.ac.kr/datasets/voxceleb/voxsrc/competition2023.html

[2] https://youtube.com/@TEDx

25 ms and hop size of 10 ms. For these systems, we performed online feature extraction and data augmentation with probability 0.6. We split the training process into two subprocesses : training and large margin fine-tuning.

For the training subprocess, we sample 2 s segment from each utterance to construct the training batch. SGD is used as the optimizer with initial learning rate 0.1 and final learning rate 0.00005, and Additive Angular Margin Loss with scale 32 as the loss function.

For the large margin fine-tuning subprocess, we removed Speed augmentation and trained with 4 s (Track 1) and 6 s (Track 2) segments. The value of margin was set to 0.3 and the scale was set to 30.

Track's 1 system was trained for 100 epochs and fine-tuned for 20 epochs, whereas Track's 2 system was trained for 60 epochs and fine-tuned for 5 epochs (our initial idea was to train this system for 150 epochs and fine-tune it for 50 epochs, however, we had to cut down the training process due to deadline). In both systems, the batch size was set to 128.

3.1.2. Scoring

For both ResNet systems, cosine distance is used to measure the distance between enrollment and test utterances. We also perform adaptive score normalization (as-norm). The imposted cohort is estimated from the VoxCeleb2-dev set by averaging the embedding of each training speaker and the cohort size is set to 300.

3.2. Offline Systems

For each track we developed an offline system that could provide complementary information, and increase the discriminative power of a system fusion. Both complementary systems are based on a Time Delay Neural Network (TDNN) architecture that is well-known in the Speaker Recognition literature [9]. The choice of this architecture was motivated by the good and consistent performance that these models have shown in speaker recognition tasks and its relatively small number of parameters, which could be an important factor given the limited amount of training data and time. The TDNNs were implemented using TensorFlow [10].

3.2.1. Architecture and Training

Our TDNN architecture is summarized in Table 1. Note that our architecture differ in some implementation details, in line with [11], with respect to the original implementation in [9, 12].

Table 1: *Complementary systems TDNN architecture. T denotes the number of input frames. (Number of parameters in the first layer assumes 24-dimensional feature vectors.)*

Layer Type	Filter/Stride	Output	Params
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	64K
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	1.313M
Conv1D-batchnorm-ReLU	$7 \times 1/1 \times 1$	$T \times 512$	1.837M
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Dense-batchnorm-ReLU	-	$T \times 1500$	775.5K
Stats Pooling (mean+stddev)	-	3000	-
Dense-batchnorm-ReLU	-	512	1.538M
Dense-batchnorm-ReLU	-	512	264.7K
Softmax	-	2	1026
Total			6.06M

TDNN systems use as input features, 24-dimensional Mel Frequency Cepstral Coefficients (MFCCs) computed from 30 filters between 20 and 7900 Hz and using a 25 ms window with 15 ms overlap. No feature normalization was used and Energy-based Voice Activity Detection (VAD) was aplied. All data augmentation tecniques in Section 2.2 were aplied offline, except specAugment which was online. For Track 2 BVSelf-VoxCeleb was also used with correspondient data augmentation.

Training was performed using SGD as optimizer with an exponentially decaying learning rate with initial value of 0.005 and Additive Margin Loss with scale 24 and 0.15 margin as the loss function. Mini-batches were constructed so that they contained 64 utterances (one speaker each). For each epoch a random length of frames between 200 and 500 is selected to construct the training batch. Training stopped when validation loss did not improved for 10 epochs. Track 1 TDNN trained for 73 epochs and Track 2 TDNN only for 25 epochs due to deadline.

3.2.2. Scoring

TDNN back-end follows a classical LDA-PLDA scoring scheme:

- Embeddings are projected to unit length, centered and whitened.
- Linear discriminant analysis (LDA) is used to project the embeddings to a lower dimension. (From 512 to 150 in our case.)
- The segments are scored using PLDA.

Both LDA and PLDA are trained on VoxCeleb2-dev. No score normalization was applied.

3.3. Quality Measurement Function

As proved by the previous edition winner [3], the use of Quality Measures to re-score verification scores can provide a huge perfomance increase, especially on VoxCeleb-like datasets. We applied the same quality measures, being the following ones:

- $\mathbf{q_1}$: speech length of the enrollment file.
- $\mathbf{q_2}$: speech length of trial file.
- $\mathbf{q_3}$: logarithm (base 10) of sum of enrollment and trial files speech length.
- $\mathbf{q_4}$: logarithm (base 10) of sum of enrollment and trial files total length.
- $\mathbf{q_5}$: SNR (signal to noise ratio) of enrollment file.
- $\mathbf{q_6}$: SNR (signal to noise ratio) of trial file.
- $\mathbf{q_7}$: NISQA [13] MOS (Mean Opinion Score) of enrollment file.
- $\mathbf{q_8}$: NISQA MOS of trial file.

For Track 1, only the six first quality measures were used, as q_7 and q_8 are based on a Deep CNN-Self-Attention Model trained on external datasets labeled with human opinion scores. For Track 2, all quality measures are used to re-score the original scores.

4. System Description Track 4

Our aim in this track is to test and fusion several diarization systems created for previous editions and other challenges to determine if combination can improve the results for this track.

4.1. Diarization systems

We make use of three diarization systems in this track:

- The **BV system** used for this track is the one created for our submission in the Albayzin-RTVE 2020 Speaker Diarization and Identity Assignment Challenge [14].

  As a summary, it extracts 23-dimensional MelFrequency Cepstral Coefficients (MFCCs) in 25 ms windows with 15 ms overlap, then VAD is performed and speaker embeddings are extracted in a 3 s window, 1.5 s hop using a TDNN. The embeddings are clustered using AHC with single linkage. Finally, a second diarization stage is performed to obtain the final diarization result.
- The **wespeaker [8] baseline diarization system**. It consists of three parts: VAD model pretrained by Silero [15], a ResNet34 embedding extractor trained with VoxCeleb and spectral clustering for clustering.
- The **Pyannote diarization** model 2.1 [16], consisting of two modules: an end-to-end speaker segmentation model, which encompasses the tasks of voice activity detection, speaker change detection and overlapped speech detection and a resegmentation module which assigns overlapped speech regions to the right speakers. This two modules were trained using DIHARD 3 [17], VoxConverse [18] and AMI [19] datasets.

  For validation we use the VoxConverse 0.3 dev subset.

5. Evaluation Protocol

The performance of the systems for Track 1 and Track 2 was evaluated using two metrics:

- The minimum detection cost function used by the NIST SRE [20] with parameters P_{target} = 0.05, C_{Miss} = 1 and $C_{FalseAlarm}$ = 1. Detection cost is the primary metric for the challenge.
- The Equal Error Rate (EER) which shows where False Acceptance (FA) and False Rejection (FR) error rates are equal.

For Track 4 system's performance evaluation was conducted using two following metrics:

- Diarisation Error Rate (DER): isthe sum of speaker error, false alarm speech and missed speech. We used a 0.25-second forgiving collar, and overlapping speech was not disregarded.
- Jaccard Error Rate (JER). Based on the Jaccard index, which is defined as the ratio between the intersection and union of two segmentations. It is computed as a 1 minus the average of Jaccard index of optimal mappings between reference and system speakers [21].

6. Fusion Scheme and Results

6.1. Track 1&2

The final output of our submitted systems for Track1&2 is an implementation of a linear fusion of the scores for all the models trained for each track and quality measures values. To find the weights of each model we use the COBYLA[22] optimizer on a 100000 balanced trials file generated from VoxCeleb2-dev, providing the optimizer with some manually computed initial weights based on the model performance and treating all quality measurements equally. The trial score was obtained according to:

$$S' = [w_1\ w_2\ ...\ w_n] \begin{bmatrix} S_1 \\ S_2 \\ ... \\ S_n \end{bmatrix} + [y_1\ y_2\ ...\ y_m] \begin{bmatrix} q_1 \\ q_2 \\ ... \\ q_m \end{bmatrix}$$

where w_i are the weights associated with each model S_i and y_i are the weights associated with each quality measure q_i.

Tables 2, 3 summarize the results obtained on the VoxSRC2023 eval and test subsets respectively. Take into account that the systems marked with an asterisk ($*$) are the ones in which Quality Measurement Functions have been applied.

Table 2: *Performance Metrics Tracks 1 & 2 eval set*

System	minDCF10	EER [%]
A: resnet34 Track1	0.2176	3.942
B: resnet34 Track2	0.3004	5.837
C: TDNN Track1	0.506	8.711
D: TDNN Track2	0.5548	10.053
A+B	0.2316	4.137
A+C	0.2122	3.973
A+B+C+D	0.2247	4.118
A*	0.2161	3.86
(A+B)*	0.2096	3.726
(A+C)*	**0.2076**	**3.708**
(A+B+C+D)*	**0.2074**	**3.697**

Table 3: *Performance Metrics Tracks 1 & 2 test set*

System	minDCF10	EER [%]
A	0.263	4.655
A*	0.2657	4.654
A+B+C+D	0.267	4.941
(A+C)*	**0.2437**	**4.394**
(A+B+C+D)*	**0.2337**	**4.262**

6.2. Track 4

To check whether the fusion of systems can lead to a perfomance boost, we use DOVER-LAP [23] to fuse together the diarization results of the different systems. Table 4 summarizes the results obtained on the the VoxConverse 0.3 dev subset and VoxSRC2023 test subset respectively.

Table 4: *Performance Metrics Track 4*

	eval Set		test Set	
Model	DER [%]	JER [%]	DER [%]	JER [%]
A: primary system	13.12	26.6	18.53	35.87
B: wespeaker baseline	7.01	**25.00**	8.31	37.52
C:pyannote	**3.84**	26.37	**7.11**	**39.85**
A+B+C	6.69	25.98	10.81	37.03
B+C	4.11	25.48	8.1633	39.21

7. References

[1] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[2] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[3] A. A. I. Y. Rostislav Makarov, Nikita Torgashov and A. Okhotnikov, "Id rd system description to voxceleb speaker recognition challenge 2022," 2022.

[4] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015.

[5] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[6] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech 2015*, 2015, pp. 3586–3589.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*. ISCA, sep 2019. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680

[8] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," 2022.

[9] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[11] Y. Liu, L. He, and J. Liu, "Large Margin Softmax Loss for Speaker Verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2357

[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 *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5796–5800.

[13] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Interspeech 2021*. ISCA, aug 2021. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2021-299

[14] R. Font and T. Grau, "The Biometric Vox System for the Albayzin-RTVE 2020 Speaker Diarization and Identity Assignment Challenge," in *Proc. IberSPEECH 2021*, 2021, pp. 86–89.

[15] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier," https://github.com/snakers4/silero-vad, 2021.

[16] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, Brno, Czech Republic, August 2021.

[17] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third dihard diarization challenge," 2021.

[18] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," 2020.

[19] I. Mccowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska Masson, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," *Int'l. Conf. on Methods and Techniques in Behavioral Research*, 01 2005.

[20] S. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2018 nist speaker recognition evaluation," 09 2019, pp. 1483–1487.

[21] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second dihard diarization challenge: Dataset, task, and baselines," 2019.

[22] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:118045691

[23] D.Raj, P.Garcia, Z.Huang, S.Watanabe, D.Povey, A.Stolcke, and S.Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021.