INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

The ID R&D System Description for Short-duration Speaker Verification Challenge 2021

Alexander Alenin[1], *Anton Okhotnikov*[1,2], *Rostislav Makarov*[1], *Nikita Torgashov*[1], *Ilya Shigabeev*[1], *Konstantin Simonchik*[1]

[1]ID R&D Inc., New York, USA
[2]SPbU, Saint Petersburg State University, Saint Petersburg, Russia
{alenin,ohotnikov,makarov,torgashov,shigabeev,simonchik}@idrnd.net

Abstract

This paper describes ID R&D team submission to the text-independent task of the Short-duration Speaker Verification (SdSV) Challenge 2021. The top performed system is a fusion of 9 Convolutional Neural Networks based on the ResNet architecture. Experiments' results of optimal NN architecture search are shown. We also present and investigate the subnetwork approach to solve the auxiliary tasks such as gender or language detection. Verification scores refinement step using quality measurements of a trial pair allowed to further minimize the target metrics. A comparative analysis of all systems used in the fusion has been provided on the VoxCeleb-1 test set, SdSV-2021 development and evaluation sets. The final submission achieves **0.69**% EER and **0.0319** minDCF on the challenge evaluation set.

Index Terms: Speaker recognition, Speaker verification, cross-lingual speaker verification, SdSV Challenge 2021

1. Introduction

The SdSV [1] is a challenge evaluating new technologies for text-dependent and text-independent speaker verification in short-duration audio conditions. The challenge is organized as two independent tasks, where Task 1 comprises speaker verification in text-dependent mode, and Task 2 in text-independent mode accordingly. This paper focuses on the Task 2 only.

Speaker verification consists of two phases - an enrollment phase and a verification phase. During the enrollment phase, the speaker's voice is recorded few times. The verification phase's objective is to automatically determine whether a test recording from an unknown speaker belongs to the enrollment speaker or not. The SdSV Challenge 2021 conducts a speaker verification with varying phonetic properties between enrollment and test utterances, and this leads to a testing protocol with monolingual and cross-lingual comparisons. Also, the length of test utterances varies from 1 to 8 seconds only, which makes speaker recognition more challenging.

2. System Setup

2.1. Input features

For training, fixed-length 2-second audio segments were used. We randomly cropped segments from each utterance in the training dataset. Then, 80-dimensional Mel filter bank log-energies with a 25 ms frame length and 10 ms step were extracted with an FFT size of 1024 over the 20-7600 Hz frequency limits. To test the models, we used 4-second input segments. A small voice activity detection module was used during both training and testing stages.

2.2. Architectures

All the systems in our submission are based on the residual neural networks [2], which made a breakthrough in the task of image classification by using very deep models, and recently have been efficiently applied to the speaker recognition task [3], [4]. A ResNet-34 architecture described in [4] was selected as our baseline system. Since the deeper models usually show a performance improvement in various tasks, we decided to apply some modifications to the baseline architecture. In particular, to increase the capacity of models we have run a series of experiments and optimised such hyperparameters as a number of residual blocks and a number of filters in each residual block. In the end, 7 modifications of the ResNet-34 model with 48, 56, 76, 82, 100 (two modifications) and 130 hidden layers were selected. Detailed architectures are shown in the Table 1.

2.3. Subnetwork Approach

To fine-tune the neural network and to solve auxiliary tasks, such as gender or language detection, we used a subnetwork approach. The subnetwork is a small neural network that leverages the performance of the main backbone: it uses the outputs from each of ResBlocks, its first convolution layer and the embedding layer (see Figure 1). The architecture of a subnetwork is the following: we apply a downsampling convolution to each ResBlock output to get the number of filters we want, then we apply a 3x3 convolution and average pooling to decrease the feature map shape and concatenate it with the next one. During the subnetwork training, the main backbone is frozen. This approach benefits from using the informatively rich features of the main backbone and allows to get small (300K - 2M parameters) and robust network.

2.4. Loss function

All our models were trained using the Additive Margin Softmax (AM-Softmax) loss function [6]. The main aim of this loss is to reduce the interclass variance by introducing the margin penalty to the target class logit. AM-Softmax showed itself as an effective loss function in face recognition and has been successfully applied to speaker recognition task as well. According to [4], the margin value was set to 0.3 and a scale value was set to 40.

3. Experiments

3.1. Datasets

For our experiments all allowed training data was used: VoxCeleb2-dev (5994 speakers) [7], VoxCeleb1-dev (1211 speakers) [8], Librispeech (*train-other-500* subset - 1166 speak-

http://dx.doi.org/10.21437/Interspeech.2021-1553

Table 1: *Models architectures*

Layer name	Output (C × F × T)	ResNet-48mnw	ResNet-56m	ResNet-76mnw	ResNet-82m	ResNet-100m	ResNet-100mnw	ResNet-130m
Conv2D	128 (96) × 80 × T	96, 3×3, stride=1	128, 3×3, stride=1	96, 3×3, stride=1	128, 3×3, stride=1	128, 3×3, stride=1	96, 3×3, stride=1	128, 3×3, stride=1
ResBlock-1	C × 80 × T	$\begin{bmatrix}3\times3,96\\3\times3,96\end{bmatrix}\times6$	$\begin{bmatrix}3\times3,96\\3\times3,96\end{bmatrix}\times6$	$\begin{bmatrix}3\times3,96\\3\times3,96\end{bmatrix}\times10$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times8$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times6$	$\begin{bmatrix}3\times3,96\\3\times3,96\end{bmatrix}\times12$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times8$
ResBlock-2	C × 40 × T/2	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times8$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times8$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times14$	$\begin{bmatrix}3\times3,160\\3\times3,160\end{bmatrix}\times12$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times16$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times16$	$\begin{bmatrix}3\times3,196\\3\times3,196\end{bmatrix}\times20$
ResBlock-3	C × 20 × T/4	$\begin{bmatrix}3\times3,160\\3\times3,160\end{bmatrix}\times6$	$\begin{bmatrix}3\times3,196\\3\times3,196\end{bmatrix}\times10$	$\begin{bmatrix}3\times3,160\\3\times3,160\end{bmatrix}\times10$	$\begin{bmatrix}3\times3,196\\3\times3,196\end{bmatrix}\times16$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times24$	$\begin{bmatrix}3\times3,160\\3\times3,160\end{bmatrix}\times18$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times32$
ResBlock-4	256 × 10 × T/8	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times4$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times4$
Flatten (C, F)	2560 × T/8	—						
Pooling	5120	ASP [5]	StatsPooling	ASP [5]	StatsPooling	StatsPooling	ASP [5]	StatsPooling
Dense	256	—						
AM-Softmax	Num. of speakers	—						

ers) [9], Mozilla Common Voice (MCV) (*Farsi* subset - 835 speakers) [10], and DeepMine train part (538 spks) [11].

The VoxCeleb datasets are multi-lingual, most of the speakers are English-speaking. The LibriSpeech is a corpus of read English speech. The MCV Farsi is a crowd-sourced set of Persian language. The DeepMine is a multi-lingual dataset including English and Persian languages.

We filtered the MCV dataset and saved speakers with more than 20 utterances, and set a maximum number of used utterances for any speaker to 200. Also, we used 50 speakers from the DeepMine training dataset as a validation subset, and this resulted in 9744 total training speakers for all datasets combined.

This year's challenge provides a development set for evaluation of the systems before submitting them to the leaderboard, however, the testing protocol corresponding to this dataset has a small number of comparisons (7071) and also has an overlap with the main evaluation dataset. Hence we created an extended development set based on the DeepMine training data. We have selected 50 speakers with the smallest amount of utterances and used them for best checkpoint selection while neural networks training and for fusion weights optimization.

As test sets, we used VoxCeleb1-test [8], DeepMine development and DeepMine extended development sets (selected from DeepMine training set).

3.2. Data augmentation

To augment training datasets, we used MUSAN corpus [12] and real room impulse responses (RIRs) database [13]. We precomputed various augmentations for each of the original datasets and expanded the size of our training set by 5 times. For each training utterance we applied 4 different augmentation strategies [4]:

- **Music**: A single music file was randomly selected from MUSAN and added to the original audio (5-15dB SNR). The duration of additive noise was matched to the duration of the original signal.
- **Noise**: Randomly selected noise from MUSAN was added to the original recording (0-15dB SNR).
- **Speech**: Three to seven speakers were randomly picked, summed together, and then added to the original signal (13-20dB SNR).
- **Reverb**: Artificially reverberated a signal via convolution with real RIRs.

3.3. Implementation details

All the described models were trained using TensorFlow 2 framework [14]. To train very deep ResNet architectures faster, Google Cloud TPUs were used. To train the models we used the following three-stage scheme:

Pre-training. We started with a model pre-training on the VoxCeleb2-dev dataset. The training strategy described in section 5 of the paper [4] was used. We trained a model with an exponentially decaying learning rate scheduler (decay steps = 10K, decay rate = 0.5) with 50K plateau steps at the beginning of training with an initial learning rate (LR) of 0.01. Each model was pre-trained for 40 epochs (5K steps per epoch) with a batch size of 256.

Fine-tuning. Then the pre-trained weights from previous stage were used for fine-tuning on the mixture of datasets: VoxCeleb2-dev, VoxCeleb1-dev, LibriSpeech (*train-other-500* subset), MCV (*Farsi*) and DeepMine. We used a LR warm-up during the first epoch (from $1e^{-6}$ to $5e^{-2}$) and then applied an exponential decay of LR every 10K steps with a rate of 0.5 for 30 epochs. This stage is defined as a **Fine-tune A** in the Table 2. During the fine-tuning stage, we noticed that increasing the frequency of DeepMine examples in training batches improved the target metrics on the development set, while showed a small degradation on the VoxCeleb1-test set. Therefore we define a **Fine-tune B** stage, which is identical to Fine-Tune A with the difference of increased proportion of DeepMine speakers in training batches.

Subnetwork training. For some of our networks we trained a subnetwork described in section 2.3 as an additional stage. The subnetworks with 2M parameters were trained on DeepMine, MCV (*Farsi*) and LibriSpeech (*train-other-500*) datasets. We used a constant LR $5e^{-2}$ for the first 4 epochs and then applied an exponential decay every 8 epochs with a rate of 0.5 for 32 consecutive epochs with 500 steps per epoch and batch size 256. The final embedding was constructed as a concatenation of a subnetwork and a backbone embedding. Comparing to the pre-trained on VoxCeleb2-dev dataset model, the subnetwork shows a 30-40% relative reduction in target metrics, but applying the same technique to the fine-tuned model doesn't provide considerable improvements (see Table 2).

At each training stage, the L2-regularization of the model's weights has been set to $5e^{-4}$. We also applied SpecAugment [15] to the input log Mel-spectrograms and randomly masked 0 to 5 frames in the time domain and 0 to 8 frequency bins with a probability of 20%. After finishing the training stage, the AM-

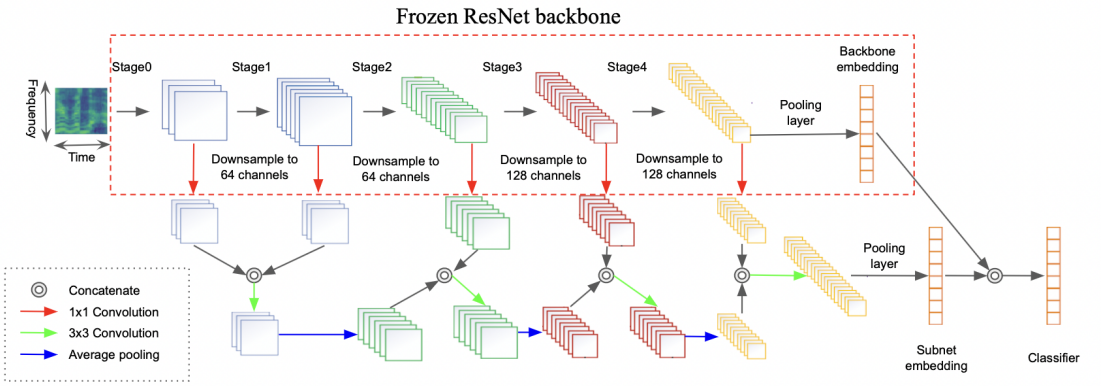

Figure 1: *Subnetwork architecture. We use different colours to indicate different resolutions in the time-frequency domain.*

Softmax layer is removed and a dense layer output is used as a system's embedding.

3.4. Backends

Extracted speaker's embeddings were scored with two different backends - cosine similarity and PLDA backend [16]. The cosine similarity backend has shown accurate results due to the usage of AM-Softmax loss function together with fine-tuning on the in-domain dataset. For PLDA model training we used Kaldi Toolbox [17]: embeddings were centered and length-normalized, and a single E-M iteration was used for PLDA parameters estimation on the Task 2 training set. Then, all verification scores were subject to score normalization and calibration.

3.5. Score normalization

We utilized an Adaptive S-Norm (AS-Norm) [18] for normalization of cosine and PLDA scores. The imposter cohort included 6000 utterances in English language from the train in-domain dataset. The addition of Farsi utterances to the cohort negatively affected the scores of cross-lingual trials, hence they were discarded. For each enrollment model and a test utterance in a trial pair, we selected the top 400 scores from the cohort to compute mean and standard deviation (normalization parameters). Then, every trial score was normalized by subtracting the computed mean and dividing it by standard deviation.

3.6. Quality Measurement Functions

To further improve calibration performance, we have used the quality measurements of files to penalize the verification scores [19]. In particular, we have considered several factors: *gender* and *language* of utterance, *length of testing audio file* [20], and a *number of files in enrollment model*. We refer to such factors as Quality Measurement Functions (QMF), and corresponding meta information could be extracted explicitly or by training an auxiliary model using the provided training data. QMFs help to normalize target and impostor distributions and thresholds for various trial conditions and have eventually improved our metrics on both - challenge dev and eval datasets. The optimal weight for each QMF factor was estimated at the final stage of a linear fusion [21].

3.6.1. Language penalty

To detect the language of a test utterance, we trained a small subnetwork (300K parameters) over the ResNet-48mnw model using the MCV (*Farsi*) and LibriSpeech datasets. We corrected a score S with a penalty term as follows: $S^* = S + C_{Lang} \cdot P_{Lang}/10$, where S is a score before applying the penalty, C_{Lang} is a penalizing constant calculating in section 3.7, and P_{Lang} is a language probability from 0 (English) to 1 (Farsi).

3.6.2. Gender penalty

We trained a small subnetwork to detect the gender of an utterance on the DeepMine train dataset in a similar fashion to the language classifier. A positive bias was added to the female trials as follow: $S^* = S + C_{Gender} \cdot P_{Gender}/10$

3.6.3. Testing file length penalty

Short-duration trials contain a proportionally smaller amount of speech, hence the system output is less confident. To deal with this problem we implemented an idea to penalize the scores of such trials more comparing to long-duration files. The correcting term was calculated as $S^* = S - C_{Len} \cdot L/100$, where L is a speech length of the testing file in seconds limited to 8.

3.6.4. Enrollment files amount penalty

Similarly to the testing file length, we used information about the enrollment models. Instead of using the length of audio, we counted the number of files for the enrollment model, which is proportional to the speech length. We computed a correcting term as follows: $S^* = S - C_N \cdot N/100$, where N is a number of enrollment files.

3.7. Fusion

The final score-level fused verification score was a linear combination of all cosine, PLDA, cosine-ASNorm, PLDA-ASNorm and QMF outputs. We utilized the COBYLA engine [22] from scikit-learn to estimate fusion weights, and optimized target metrics on extended development set described in section 3.1.

3.8. Evaluation

For the evaluation of the system's performance two metrics were used:

- The minimum of detection cost function (minDCF) used by the NIST SRE [24] with parameters $C_{Miss} = 10$, $C_{FalseAlarm} = 1$, and $P_{Target} = 0.01$.

Table 2: *Results on the VoxCeleb1-test, SdSV-dev and SdSV-eval sets*

Model	**Stage**	**VoxCeleb1-test** EER [%]	**SdSV dev** EER [%]	**SdSV dev** minDCF	**SdSV eval(LB)** EER [%]	**SdSV eval(LB)** minDCF
X-Vector Baseline [23]	-	-	-	-	10.65	0.4318
ResNet-48mnw	Pre-training	1.40	3.27	0.1171	2.44	0.1139
	Fine-tune B	1.65	1.77	0.0730	1.23	0.0561
ResNet-56m	Pre-training	1.39	3.21	0.1186	2.45	0.1143
	Fine-tune A	1.48	1.71	0.0623	1.27	0.0562
	Subnetwork	2.47	1.77	0.0693	1.24	0.0530
ResNet-76mnw	Pre-training	1.28	3.07	0.1099	2.27	0.1060
	Fine-tune A	1.28	1.57	0.0700	1.26	0.0580
	Subnetwork	2.84	1.56	0.0727	1.18	0.0540
ResNet-82m	Pre-training	1.24	2.59	0.1095	2.31	0.1063
	Fine-tune A	1.32	1.29	0.0695	1.19	0.0542
	Subnetwork	2.06	1.45	0.0642	1.18	0.0502
ResNet-100m	Pre-training	1.03	2.73	0.1155	2.28	0.1055
	Fine-tune A	1.16	1.30	0.0648	1.13	0.0517
	Subnetwork	2.22	1.36	0.0698	1.13	0.0506
	Fine-tune B	1.32	1.43	0.0633	**1.11**	**0.0498**
ResNet-100mnw	Pre-training	1.04	2.86	0.1171	2.28	0.1071
	Fine-tune A	1.11	1.85	0.0758	1.34	0.0634
ResNet-130m	Pre-training	0.94	2.86	0.1174	2.29	0.1058
	Fine-tune B	1.04	1.35	0.0652	1.14	0.0527
Fusion A: cosine only	-	-	0.89	0.0415	0.83	0.0372
Fusion B: A + plda + ASNorm	-	-	0.82	0.0386	0.79	0.0353
Final Fusion: B + QMF	-	-	**0.67**	**0.0282**	**0.69**	**0.0319**

Table 3: *Detailed results of the final submission*

Condition	**EER, %**	**MinDCF**
Overall Results	**0.69**	**0.0319**
Male	0.40	0.0240
Female	0.88	0.0360
English	0.65	0.0326
Farsi	0.72	0.0310
English-Male	0.41	0.0254
English-Female	0.79	0.0363
Farsi-Male	0.38	0.0230
Farsi-Female	0.94	0.0351

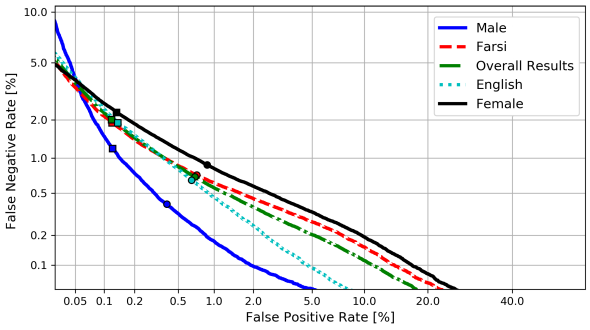

Figure 2: *DET plots of the final submission*

- The Equal Error Rate (EER), which corresponds to the operating point of equal False Acceptance and False Rejection error rates.

4. Results

The testing results on the VoxCeleb1-test, SdSV-dev and SdSV-eval datasets are presented in the Table 2. For single models, we show the results of a cosine similarity backend as a scoring function. The best performance on the validation dataset is achieved by the fusion of 9 systems (stages highlighted in bold in the Table 2), where the output verification score is a weighted average of each system scores, followed by the score normalization and a QMF refinement. The final metrics of our best fusion on the evaluation set are **0.69**% EER and **0.0319** minDCF.

This year's challenge provides a detailed output for each submission, and our final submission results are presented in the Table 3 and on Figure 2. According to these results, our system can perform well for both English and Farsi languages. There is also a strong bias between male and female trials, which we think is due to the predominance of female comparisons in the evaluation protocol: trained gender detector showed that female trials exceed male comparisons by a factor of 1.6.

The results in Table 2 show that a single ResNet-48mnw (pre-training stage) with cosine backend outperforms the challenge baseline with PLDA backend by a factor of 4. We can also see that proposed subnetwork training allowed to further improve target metrics given a small amount of tuning data, however we expect that whole neural network fine-tuning would outperform this approach if a lot of data is given. Table 2 also demonstrates that usage of multi-factor QMF calibration can significantly reduce minDCF and EER of a system.

5. Conclusions

The proposed modification of the baseline ResNet-34 [4] architecture allows to obtain a high-accurate single-model system and achieve decent performance on the challenge evaluation set using the weighted fusion of multiple frontend and backend models. Experiments with various depth and width of the ResNet architecture have shown that the best performing architectures are the ones that have a higher number of filters in earlier stages (ResBlocks). We also noted that deeper models can provide more accurate results. Also, the fine-tuning on the challenge training data allowed us to significantly reduce EER and minDCF on validation and evaluation sets.

6. References

[1] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan," *arXiv preprint arXiv:1912.06311*, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[4] D. Garcia-Romero, G. Sell, and A. Mccree, "MagNetO: X-vector Magnitude Estimation Network plus Offset for Improved Speaker Recognition," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 1–8. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-1

[5] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[6] 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.

[7] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[8] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[9] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[10] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[11] H. Zeinali, L. Burget, and J. H. Černockỳ, "A multi purpose and large scale speech corpus in persian and english for speaker and speech recognition: the deepmine database," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 397–402.

[12] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[13] I. Szoke, M. Skacel, L. Mosner, J. Paliesek, and J. Cernocky, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, p. 863–876, Aug 2019. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2019.2917582

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[15] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[16] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel, "Text-dependent speaker recognition using plda with uncertainty propagation," *matrix*, vol. 500, no. 1, 2013.

[17] 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.

[18] S.-C. Yin, R. Rose, and P. Kenny, "Adaptive score normalization for progressive model adaptation in text independent speaker verification," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4857–4860.

[19] J. Fierrez-Aguilar, J. Ortega-Garcia, J. Gonzalez-Rodriguez, and J. Bigun, "Discriminative multimodal biometric authentication based on quality measures," *Pattern recognition*, vol. 38, no. 5, pp. 777–779, 2005.

[20] M. I. Mandasari, R. Saeidi, M. McLaren, and D. A. van Leeuwen, "Quality measure functions for calibration of speaker recognition systems in various duration conditions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2425–2438, 2013.

[21] K. Nandakumar, Y. Chen, A. K. Jain, and S. C. Dass, "Quality-based score level fusion in multibiometric systems," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4. IEEE, 2006, pp. 473–476.

[22] M. J. Powell, "A view of algorithms for optimization without derivatives," *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, vol. 43, no. 5, pp. 170–174, 2007.

[23] 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)*. IEEE, 2019, pp. 5796–5800.

[24] Nist 2018 speaker recognition evaluation plan. [Online]. Available: https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf