STC-innovation Far-Field Speaker Verification Challenge 2020 System Description

Aleksei Gusev[1,2]*, Vladimir Volokhov*[2]*, Alisa Vinogradova*[1,2]*, Tseren Andzhukaev*[2]*, Andrey Shulipa*[1]*, Sergey Novoselov*[1,2]*, Timur Pekhovsky*[2]*, Alexander Kozlov*[2]

[1]ITMO University, St. Petersburg, Russia
[2]STC-Innovations Ltd., St. Petersburg, Russia

{gusev-a, volokhov, gazizullina, andzhukaev, shulipa, novoselov, tim, kozlov-a}@speechpro.com

Abstract

This paper presents speaker recognition (SR) systems submitted by the Speech Technology Center (STC) team to the Far-Field Speaker Verification Challenge 2020. SR tasks of the challenge are focused on the problem of far-field text-dependent speaker verification from single microphone array (Track 1), far-field text-independent speaker verification from single microphone array (Track 2) and far-field text-dependent speaker verification from distributed microphone arrays (Track 3).

In this paper, we present techniques and ideas underlying our best performing models. A number of experiments on x-vector-based and ResNet-like architectures show that ResNet topology based networks outperform x-vector-based systems. Submitted systems are the fusions of ResNet34-based extractors, trained on 80 Log Mel-filter bank energies (MFBs) post-processed with U-net-like voice activity detector (VAD). The best systems for the Track 1, Track 2 and Track 3 achieved 5.08% EER and 0.500 C_{det}^{min}, 5.39% EER and 0.541 C_{det}^{min} and 5.53% EER and 0.458 C_{det}^{min} on the challenge evaluation sets respectively.

Index Terms: FFSVC, speaker recognition, deep neural network, domain adaptation, neural network-based VAD.

1. System components

1.1. Feature extraction

All systems presented in this paper take 80-dimensional Log Mel-filter Bank Energies extracted from 16kHz raw input signals as input features. We compute MFBs from the signal with 25ms frame-length and 15ms overlap.

Additionally, we use per-utterance Cepstral Mean Normalization (CMN) over a 3-second sliding window over the stack of MFBs to compensate for the channel effects and noise by transforming data to have zero mean [1]. The VAD was used after the CMN-normalization procedure. We further apply global mean and standard deviation (std) normalization for each utterance with the pre-computed 80-dimensional vectors of means and stds over this utterance.

1.2. Voice activity detector

In this work we explored two types of VADs for the SR task:

- energy-based VAD from the Kaldi Toolkit [2];
- neural network-based VAD.

Neural network-based VAD uses U-net architecture [3] as a backbone and is described in more detail in our papers [4, 5]. It was trained on the large-scale dataset of telephone channel audios (telephone part of data from the NIST SRE challenge and our proprietary Russian speech subcorpus RusTelecom [6]). We have found that VAD trained on telephone data produces high-quality results for the microphone channel audios, that is the reason why we did not train our VAD for microphone data from scratch.

Preprocessing of VAD input data consisted of extraction of 8kHz 23-dimensional Mel-Frequency Cepstral Coefficients (MFCC) features from the raw signal with 25ms frame-length and 20ms shift. We found 23-dimensional MFCCs to be a trade-off between the quality of embeddings extracted from these features and the speed of training and inference.

Since training and test data used for the FFSVC 2020 consists of 16kHz microphone speech, VAD markup was first extracted from the 23-dimensional MFCC features, computed for the audios down-sampled from 16kHz to 8kHz, then the resultant markup was used to extract voiced frames from the 80-dimensional MFB features calculated from the same raw waveform data.

1.3. Embedding extractors

We used two kinds of neural network architectures to process acoustic features – ResNet-based and x-vector-based systems.

ResNet-based. Table 1 describes ResNet34 [7] architecture we used. ReLU activation and batch normalization follow each convolutional layer, and Maxout activation [8] is used for the embedding layer. Statistics pooling layer aggregates features over time and spectral dimensions before the segment-level embedding layers. In the paper, we used several embedding extractors based on ResNet34 architecture. These extractors differ from each in the data used for training, the type of data augmentation, and the voice activity detector.

X-vector-based. We took extended TDNN-based x-vectors [9] as a baseline. Then we removed dilations from convolutional kernels to avoid the possibility for the griding artifacts and replaced ReLU activation with its leaky version (Table 2). Several experiments with the width and depth of the network have shown no further improvement for the number of filters beyond 512 and the number of frame-level blocks (comprised of 1-dimensional convolutional layer and 1x1 1-dimensional convolutional layer) beyond 4. Similarly, the addition of extra fully connected segment-level layers did not affect the verification accuracy. It was found that after the network reaches a certain depth, the performance tends to saturate towards mean accuracy due to the lack of local spectral and global temporal information in the intermediate layers. We tried to add temporal attention on the base of the Squeeze-and-Excitation (SE) block [10] after each convolutional block to overcome the latter limitation,

Table 1: *Architecture configuration of the embedding extractor based on ResNet34*

Layer name	**Structure**	**Output**
Input	80 MFB log-energy	80 × 200 × 1
Conv2D-1	3 × 3, stride 1	80 × 200 × 32
ResNet-1	$\begin{bmatrix}3\times3,32\\3\times3,32\end{bmatrix}\times 3$, st. 1	80 × 200 × 32
ResNet-2	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times 4$, st. 2	40 × 100 × 64
ResNet-3	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times 6$, st. 2	20 × 50 × 128
ResNet-4	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times 3$, st. 2	10 × 25 × 256
StatsPool	mean and std	20 × 256
Flatten	–	5120
Dense1	embedding layer	512
Dense2	output layer	N_{spk}

however, no significant improvement over the basic version was reached.

Table 2: *Architecture configuration of the embedding extractor based on x-vectors*

Layer name	**Layer context**	**Output**
Input	80 MFB log-energy	80 × 200
Frame1.1	$[t-2:t-2]$	512 × 200
Frame1.2	$[t]$	512 × 200
Frame2.1	$[t-1:t-1]$	512 × 200
Frame2.2	$[t]$	512 × 200
Frame3.1	$[t-1:t-1]$	512 × 200
Frame3.2	$[t]$	512 × 200
Frame4.1	$[t-1:t-1]$	512 × 200
Frame4.2	$[t]$	512 × 200
Frame5	$[t]$	1500 × 200
StatsPool	mean and std	2 × 1500
Flatten	–	3000
Dense1	embedding layer	512
Dense2	output layer	N_{spk}

1.4. Domain adaptation

In this work, we utilized different domain adaptation techniques:

- based on the addition of in-domain data to the training set to fine-tune and train embedding extractor that solves close-set speaker identification task;
- based on mean speaker embedding subtraction. The mean vector is calculated over the training FFSVC dataset;
- based on two mean speaker embedding subtraction. The main idea is to calculate two vectors of mean values over the training FFSVC dataset for the enrollment and test files independently;
- based on MultiReader adaptation technique [11], which was used at the embedding extractor training stage. The main idea is to train embedding extractor using two heads, each of which is intended to classify either a large number of speaker IDs from out-of-domain data or a small number of speaker IDs from in-domain data.

1.5. Multi-channel fusion

Trial pairs are constructed of single enrollment recording from 25cm distance cell phone and multiple test recordings from single (Track 1 and Track 2) or multiple (Track 3) far-field microphone array. The presence of multiple test files for the enrolled speaker fragment allowed us to fuse information from test utterances in several ways:

- by averaging all enrollment-test trials scores for each trial;
- by choosing the maximum score from comparison set of one enrollment embedding with all test embeddings for each trial;
- by computing average test embedding for one trial and comparing it with the associated enrollment embedding.

We found that embedding averaging works slightly better than other methods in terms of verification metrics.

1.6. Back-end scoring

Cosine similarity was chosen to be used as a back-end scoring method:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathbf{x_1}^T \mathbf{x_2}}{\|\mathbf{x_1}\| \|\mathbf{x_2}\|}, \tag{1}$$

where $(\mathbf{x_1}, \mathbf{x_2})$ are speaker embedding vectors.

2. Datasets

We used several datasets for our experiments.

Model pre-training dataset. We used concatenated VoxCeleb1 and VoxCeleb2 (SLR47) [12] corpus datasets to pre-train all our models. The overall number of speakers in the resultant set was 7146. Augmented data was generated using standard Kaldi augmentation recipe (reverberation, babble, music and noise) using the freely available MUSAN and simulated Room Impulse Response (RIR) datasets[1].

Model fine-tuning dataset. We have expanded model pre-training data with additional 2099 speakers from several Chinese Mandarin corpora (SLR33, SLR62, SLR82, SLR85, FFSVC train set) to add domain knowledge. We concatenated multiple short-duration utterances of one speaker into multiple larger files of 20 sec for training convenience. SLR33 and SLR62 sets were augmented similarly to VoxCeleb sets, whereas SLR82 and SLR85 relatively noisy sets were augmented only by reverberation. Both augmented and non-augmented versions of the FFSVC train set were used in our experiments. During the construction of the extended dataset speaker overlaps between different datasets were take into account.

3. Experiments

3.1. Pre-training of embedding extractor

Both ResNet and x-vector-based embedding extractor models were trained on model pre-training dataset from Section 2 We performed training of our models using batches, consisting of randomly sampled sequence of 200 MFB features (2s of speech). AM-Softmax loss function [13] was taken for objective (with optimal margin and scale parameter settings fixed to 0.2 and 30 respectively). For ResNet-based models we used Adam optimizer with the starting learning rate fixed to 0.001

[1] http://www.openslr.org

and divide it by ten every 2 epochs. In x-vector training, SGD demonstrated better convergence in the combination with cyclic learning rate scheduling policy with the minimum and maximum learning rate parameters set to 0.002 and 0.12 respectively. During one epoch the full pass of train data was done.

3.2. Fine-tuning of embedding extractors

Fine-tuning of ResNet-based extractors was done in two steps. First, we trained segment-level layers and newly initialized classification head, with convolutional layers frozen. Second, we unfroze convolutional layers and re-trained the overall network with a low learning rate. We used the FFSVC development set and original VoxCeleb 1 test set for model validation in this task.

The MultiReader adaptation technique [11] was used as an alternative fine-tuning approach. We trained embedding extractor based on ResNet34 architecture using two heads. The first head is used to classify a large number of speaker IDs from out-of-domain data (7146 speakers from VoxCeleb1 and VoxCeleb2 datasets). The second head is used to classify a small number of speaker IDs from in-domain data (120 speakers from the FFSVC train set). Since the in-domain dataset does not contain enough amount of data, training the embedding extractor on it can lead to overfitting. Requiring the embedding extractor to perform reasonably well also on out-of-domain data helps to regularize the embedding extractor. We tried to minimize the following cost function based on AM-Softmax:

$$\mathcal{L}(D; \mathbf{W}) = 0.5 \cdot \mathcal{L}(D_1; \mathbf{W_1}) + 0.5 \cdot \mathcal{L}(D_2; \mathbf{W_2}), \qquad (2)$$

where D_1 and D_2 are out-of-domain and in-domain data correspondingly, $\mathbf{W_1}$ and $\mathbf{W_2}$ are model parameter sets that allow to compute outputs for the first and second heads correspondingly, $D = D_1 \cup D_2$, $\mathbf{W} = \mathbf{W_1} \cup \mathbf{W_2}$.

We used a single-headed ResNet34 model [4] trained on original and augmented VoxCeleb1 and VoxCeleb2 datasets as the initialization for frame-level, segment-level and the first output layers of two-headed ResNet34 model. The first layer and the frame level of the two-headed ResNet34 model were frozen at the beginning of the MultiReader adaptation procedure. Layer freezing was maintained until convergence and then all layers were unfrozen and training procedure was continued with a reduced learning rate until convergence. We used the original and Kaldi augmented FFSVC train set in this case.

4. Results and discussion

Table 3 displays the results of experiments on several systems developed for the Track 2 of the challenge. The performance is measured on the FFSVC development set in terms of EER (Equal Error Rate) and C_{det}^{min} (Minimum Detection Cost).

The ResNet34-based system with energy VAD and no adaptation to the specificity of the domain was taken for the baseline. We used only concatenated VoxCeleb1 and VoxCeleb2 datasets and its augmented versions for training our baseline system. The maximum score fusion method mentioned in Subsection 1.5 was used to average information from several multi-microphone test utterances. A number of changes were done to the baseline system:

- the expansion of the training set and speaker IDs with Chinese Mandarin datasets;
- the use of U-net-like VAD in training and testing stages;
- the use of one of the adaptation methods described in Subsection 1.4;
- computation of average test embedding vector for one trial and comparison of it with the associated enrollment embedding.

Table 3: *Results of our systems on FFSVC 2020 Track 2 (development set)*

ID	System	Properties	C_{det}^{min}	EER
1	ResNet34	initial model, energy VAD, max score for test files	0.820	8.95
2	ResNet34	ID1 + neural VAD	0.712	8.23
3	ResNet34	ID2 + mean vector for test files	0.700	8.24
4	ResNet34	ID3 + more data after VAD	0.688	8.26
5	ResNet34	ID4 + mean vector substraction	0.672	8.22
6	ResNet34	ID5 + two mean vector substraction	0.655	7.47
7	ResNet34	ID4 + fine-tune model	0.562	4.85
8	ResNet34	ID7 + two mean vector substraction	0.484	4.46
9	X-vectors	Extended TDNN + energy VAD	0.890	12.30
10	ResNet34	ID1 + MultiReader, mean vector for test files	0.627	5.46

Analysis of results allows us to make the following conclusions based on Table 3:

- expansion of the training set with the in-domain Chinese Mandarin datasets allows to improve performance of verification system (ID7);
- the use of U-net-like VAD at training and testing stages allows to improve the performance of verification system. This gain is attributable to the fact that U-net-like VAD produces more accurate speech detections in the presence of distortions compared to energy-based VAD (ID2);
- adaptation by one or two mean speaker embeddings subtraction allows to improve the performance of the verification system. The two mean adaptation technique gives special improvement as it accounts for the fact that enrollment and test recordings are formed using various devices (ID6, ID8);
- the use of the MultiReader adaptation technique improves the performance of the verification system. However, this approach requires careful selection of the learning rate (ID10);
- the use of multi-channel fusion approaches gives better results than the comparison between enrollment embedding and randomly selected test embedding for one trial. We did not see much difference in the performance of the verification system for different multi-channel fusion approaches (ID2, ID3).

Table 4: *Results of our best single and fused systems on FFSVC 2020 (development set and evaluation set)*

System	Task	Dev set		Eval set	
		C_{det}^{min}	EER	C_{det}^{min}	EER
Single-1	Track 1	0.490	4.24	–	–
Fusion-1	Track 1	0.462	3.66	0.500	5.08
Single-2	Track 2	0.484	4.46	0.564	5.61
Fusion-2	Track 2	0.472	4.28	0.541	5.39
Single-3	Track 3	0.434	3.35	–	–
Fusion-3	Track 3	0.417	3.22	0.458	5.53

Table 5: *Computation resources*

System	Real time factor	Memory (MB)
ResNet34 (CPU)	14	248
ResNet34 (GPU)	233	248

We presented the best results of our single and fused verification systems for all tracks in the Table 4. We used approaches similar to Track 2 to improve the performance of the verification system for Track 1 and Track 3. However, we used only the text-dependent part of the FFSVC train set for adaptation in Track 1 and Track 3 for better performance of our systems.

5. Computation resources

Processing times were measured on a machine with an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz running Ubuntu 16.04 with the CUDA 10.1 release and equipped with NVIDIA GeForce GTX 1080 Ti. Neural networks and tensor computations were constructed using the PyTorch package of version 1.5.0. A key computational bottleneck of the SR system is an embedding extraction part. Therefore, we benchmarked our best embedding extractors based on ResNet34 in terms of CPU (single-threaded) and GPU execution times (Table 5). Table 5 also reports for the amount of memory required to process a speech fragment of two seconds.

6. Conclusions

Obtained results confirm that deep ResNet architectures are robust and allow to obtain a good quality of speaker verification for short-duration utterances. Our best performing system for FFSVC 2020 (development set) protocols is ResNet34-based system built on high-frequency resolution MFB features. It is trained with AM-Softmax-based loss function. We should also note that utilization of additional in-domain data, our U-net-like VAD, various adaptation techniques, and multi-channel fusion approaches provide additional performance gains for proposed SR systems in considered tasks.

7. References

[1] S. Furui, “Cepstral analysis technique for automatic speaker verification,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *ICASSP 2018 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada, Proceedings*, 2018, pp. 5329–5333.

[3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI 2015 – 18th International Conference on Medical Image Computing and Computer Assisted Intervention, October 5-9, Munich, Germany, Proceedings*, 2015, pp. 234–241.

[4] A. Gusev, V. Volokhov, T. Andzhukaev, S. Novoselov, G. Lavrentyeva, M. Volkova, A. Gazizullina, A. Shulipa, A. Gorlanov, A. Avdeeva, A. Ivanov, A. Kozlov, T. Pekhovsky, and Y. Matveev, “Deep speaker embeddings for far-field speaker recognition on short utterances,” in *Odyssey 2020 – The Speaker and Language Recognition Workshop, November 02-05, Tokyo, Japan, Proceedings*, 2020.

[5] G. Lavrentyeva, M. Volkova, A. Avdeeva, S. Novoselov, A. Gorlanov, T. Andzukaev, A. Ivanov, and A. Kozlov, “Blind speech signal quality estimation for speaker verification systems,” in *to appear in INTERSPEECH 2020 – 21th Annual Conference of the International Speech Communication Association, October 25-29, Shanghai, China, Proceedings*, 2020.

[6] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Shchemelinin, “On deep speaker embeddings for text-independent speaker recognition,” in *Odyssey 2018 – The Speaker and Language Recognition Workshop, June 26-29, Les Sables d’Olonne, France, Proceedings*, 2018, pp. 378–385.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR 2016 – 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 26 - July 1, Las Vegas, Nevada, USA, Proceedings*, 2016, pp. 770–778.

[8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in *ICML 2013 – 30th International Conference on Machine Learning, June 17-19, Atlanta, Georgia, USA, Proceedings*, vol. 28, no. 3, 2013, pp. 1319–1327.

[9] D. Garcia-Romero, D. Snyder, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “X-vector DNN refinement with full-length recordings for speaker recognition,” in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association, September 15-19, Graz, Austria, Proceedings*, 2019, pp. 1493–1496.

[10] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *CVPR 2018 – 2018 IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, Salt Lake City, Utah, USA, Proceedings*, 2018, pp. 7132–7141.

[11] L. Wan, Q. Wang, A. Papir, and I. Moreno, “Generalized end-to-end loss for speaker verification,” in *ICASSP 2018 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada, Proceedings*, 2018, pp. 4879–4883.

[12] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “VoxCeleb: Large-scale speaker verification in the wild,” *Computer Speech and Language*, vol. 60, p. 101027, 2020.

[13] F. Wang, W. Liu, H. Liu, and J. Cheng, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.