Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

Deep Speaker Embeddings for Far-Field Speaker Recognition on Short Utterances

Aleksei Gusev[1,2], Vladimir Volokhov[2], Tseren Andzhukaev[2], Sergey Novoselov[1,2], Galina Lavrentyeva[1,2], Marina Volkova[1,2], Alice Gazizullina[1,2], Andrey Shulipa[1], Artem Gorlanov[2], Anastasia Avdeeva[2], Artem Ivanov[2], Alexander Kozlov[2], Timur Pekhovsky[2], Yuri Matveev[1,2]

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

{gusev-a,volokhov,andzhukaev,novoselov,lavrentyeva,volkova,gazizullina,
shulipa,gorlanov,avdeeva-a,ivanov-ar,kozlov-a,tim,matveev}@speechpro.com

Abstract

Speaker recognition systems based on deep speaker embeddings have achieved significant performance in controlled conditions according to the results obtained for early NIST SRE (Speaker Recognition Evaluation) datasets. From the practical point of view, taking into account the increased interest in virtual assistants (such as Amazon Alexa, Google Home, Apple Siri, etc.), speaker verification on short utterances in uncontrolled noisy environment conditions is one of the most challenging and highly demanded tasks. This paper presents approaches aimed to achieve two goals: a) improve the quality of far-field speaker verification systems in the presence of environmental noise, reverberation and b) reduce the system quality degradation for short utterances. For these purposes, we considered deep neural network architectures based on TDNN (Time Delay Neural Network) and ResNet (Residual Neural Network) blocks. We experimented with state-of-the-art embedding extractors and their training procedures. Obtained results confirm that ResNet architectures outperform the standard x-vector approach in terms of speaker verification quality for both long-duration and short-duration utterances. We also investigate the impact of speech activity detector, different scoring models, adaptation and score normalization techniques. The experimental results are presented for publicly available data and verification protocols for the VoxCeleb1, VoxCeleb2, and VOiCES datasets.

1. Introduction

The increasing interest in reliable means of guarding and restricting access to informational resources requires development of new authentication methods. Biometric recognition remains one of the key priority research areas in this field.

Today Automatic Speaker Verification (ASV) systems are reliable, convenient, low cost and are in compliance with security regulations what makes them a subject of increased interest to state law enforcement agencies and commercial structures. Moreover, such systems can operate on different input-output devices and communication channels (landline, mobile telephone networks, IP telephony, etc.).

The latest results obtained for the telephone part of National Institute of Standards and Technology Speaker Recognition Evaluation (NIST SRE) datasets demonstrated that speaker recognition (SR) systems based on deep speaker embeddings had achieved significant results in controlled conditions [1]. However, speaker verification on short utterances is still one of the challenging tasks in the text-independent speaker recognition field.

Taking into account the increased interest in virtual assistants, the demand for far-field speaker verification on short utterances (such as wake-up words and short commands) in uncontrolled noisy environment conditions is very high. Quality of such systems depends on the presence of a number of factors, with channel mismatch, environmental noise and room reverberation being most prominent ones. This was confirmed by the VOiCES 2019 challenge [2, 3] aimed to support research in the area of SR and automatic speech recognition (ASR) with the special focus on single channel far-field audio under noisy conditions.

This paper presents approaches aimed to achieve two goals: to improve the performance of far-field speaker verification systems in the presence of environmental noise and reverberation, and to reduce the system quality degradation for short utterances. In order to accomplish this task, we explored different state-of-the-art deep neural network architectures and their applicability to speaker verification task in uncontrolled environmental conditions on publicly available data and verification protocols for the VoxCeleb1, VoxCeleb2, and VOiCES datasets.

We experimented with deep speaker embedding extractors based on TDNN (Time Delay Neural Network) [4] and ResNet (Residual Neural Network) [1, 5] blocks and different training objectives. A detailed description of the extractors is presented in Section 4. Special attention was paid to the impact of deep neural network speech activity detector presented in 3.2 which is more robust against noise and other distortions compared to classical energy-based methods. In this paper, we also analyzed different scoring models, adaptation and score normalization techniques and estimated their contribution to the final system performance.

All obtained experimental results and their comparison with the standard x-vector approach are considered in Section 5. The performance of the proposed systems is measured in terms of Equal Error Rate (EER) and Minimum Detection Cost Function (minDCF).

10.21437/Odyssey.2020-26

2. Related work

Deep learning approaches for speaker representation undoubtedly let the speaker recognition field to reach new levels of its evolution. The ongoing progress of such methods leads to implementation of new state-of-the-art SR systems.

2.1. DNN speaker embeddings

Deep neural network (DNN) based speaker embedding extractors substantially improve performance of speaker ID systems in challenging conditions. TDNN based x-vector system significantly outperforms conventional i-vector based system in terms of speaker recognition performance and hence is a new baseline for text-independent SR task [4]. The authors proposed an end-to-end system that learns to classify speakers and produce representative deep speaker embeddings able to generalize well to speakers that have not been seen in the training data. The key feature of the proposed architecture was a statistics pooling layer designed to aggregate frame-level features along the time dimension by computing their standard deviation (std) and mean. X-vectors, extracted from an intermediate layer of the neural network which comes after the statistics pooling layer, demonstrate properties similar to those of i-vectors from total variability space, which makes it possible to effectively use them in the standard Linear Discriminant Analysis (LDA) followed by Probabilistic Linear Discriminant Analysis (PLDA) [6] backend.

Studies such as [7, 8] follow this deep speaker representation direction with improvement of SR performance. For example, the system from [7] proposed by JHU team for NIST SRE 2018 used the extended version of TDNN based architecture – E-TDNN. The differences include an additional TDNN layer with wider temporal context and unit context TDNN layers between wide context TDNN layers.

Paper [8] proposed to use an alternative training objective A-Softmax (Angular Margin Softmax) activation [9] to be used instead of the standard Softmax to train a so called c-vector based system. The main characteristics of the proposed architecture were residual blocks [10] built using TDNN architecture and MFM (Max-Feature-Map) activations [11] used instead of ReLU.

2.2. Speaker embeddings for short utterances

Short utterances and far-field microphones are new challenging conditions for the SR task. Recent papers [12, 13] devoted to this problem demonstrate that substantial improvements can be achieved by deeper architectures such as residual networks [10] and by more accurate task-oriented augmentation of training data.

An analysis of the degradation of speaker verification quality on short intervals of the data from the VoxCeleb1 dataset is carried out in [12, 13]. Authors of [12] demonstrate impressive results for "in the wild" scenario. They propose a modified residual network with a NetVLAD/GhostVLAD layer for feature aggregation along the temporal axis. This layer is aimed to apply self-attentive mechanism with learnable dictionary encoding [14].

An alternative approach for feature aggregation over time in a residual network is discussed in [13]. The authors propose a simple and elegant Time-Distributed Voting (TDV) method. It demonstrates significant quality improvement with short utterances in comparison with NetVLAD solution. However, it does not perform so well with longer duration utterances.

2.3. Speaker embeddings for distant speaker recognition

Recent progress and growing popularity of virtual assistants in smart home systems and smart devices have led to higher requirements not only for speech recognition but also for the reliability of the biometric systems under far-field conditions. In 2019 the VOiCES from a Distance Challenge [3] was organised to support the research in the area of SR and ASR with the special focus on single channel distant/far-field audio under noisy conditions. The challenge was based on the freely-available VOiCES corpus [2] released several months before. Almost all systems proposed during the challenge exploited different architectures of neural networks to obtain deep speaker representations. To reduce the effects of room reverberation and various kinds of distortions, some researches use more accurate task-oriented data augmentation [15, 16, 17, 18] and speech enhancement methods [16] based on single-channel weighted prediction error (WPE) [19].

2.4. Loss function for speaker embedding learning

Over the past few years, in the face recognition field, many loss functions have been proposed for the effective training of embedding extractors: A-Softmax [9], Additive Margin Softmax (AM-Softmax) [20], Additive Angular Margin Softmax (AAM-Softmax) [21], Dissected Softmax (D-Softmax) [22] based loss functions. Recent studies in speaker verification field have demonstrated impressive performance of the AM-Softmax based training loss function for speaker ID systems [1, 5]. Thus, in this work, we are mainly focused on the well-performing AM-Softmax based loss function and additionally experiment with D-softmax loss.

AM-Softmax based loss function is defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_i \frac{e^{s\left(cos\left(\theta_{y_i}\right)-m\right)}}{e^{s\left(cos\left(\theta_{y_i}\right)-m\right)} + \sum_{j\neq y_i} e^{s\left(cos\left(\theta_j\right)\right)}}, \quad (1)$$

where $\cos\left(\theta_{y_i}\right) = \mathbf{w}_{y_i}^T \mathbf{f}_i / \left(\|\mathbf{w}_{y_i}\| \, \|\mathbf{f}_i\|\right)$, $\mathbf{w}_{y_i}$ is the weight vector of class y_i, and $\mathbf{f}_i$ is the input to the layer i. Parameter s is an adjustable scale factor and m is the penalty margin. AM-Softmax loss allows to compare speaker embeddings by cosine distance.

D-Softmax is a new loss function that was presented recently in [22] as an effective objective for face embedding learning. Authors of [22] speculate that the intra- and inter-class objectives in the Softmax loss are entangled, therefore a well-optimized inter-class objective leads to relaxation on the intra-class objective, and vice versa. The main idea of D-Softmax loss is to dissect the Softmax loss into independent intra- and inter-class objective.

D-Softmax loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{intra} + \mathcal{L}_{inter} = \\ -\frac{1}{N}\sum_i \left(\frac{e^{s\cos\left(\theta_{y_i}\right)}}{e^{s\cos\left(\theta_{y_i}\right)} + \epsilon} + \frac{1}{1 + \sum_{j\neq y_i} e^{s\cos\left(\theta_j\right)}} \right), \quad (2)$$

where ϵ and s are customizable parameters.

3. Description of the system components

3.1. Feature extraction

All our embedding extractors use MFCC (Mel Frequency Cepstral Coefficients) and MFB (Log Mel-filter Bank Energies)

from 16 kHz raw input signals (standard Kaldi recipe) as low-level features:

- 40 dimensional MFCC extracted from the raw signal with 25ms frame-length and 15ms overlap;
- 80 dimensional MFB extracted from the raw signal with 25ms frame-length and 15ms overlap.

Extracted voice features additionally undergo one of the two different post-processing steps depending on the type of embedding extractor used afterwards:

- local Cepstral Mean Normalization (CMN-normalization) over a 3-second sliding window;
- local CMN-normalization over a 3-second sliding window and global Cepstral Mean and Variance Normalization (CMVN-normalization) over the whole utterance.

To extract speech out of the audio signal, we used our own neural network-based Voice Activity Detector (VAD) system that was trained on 8kHz telephone and a small amount of microphone data. Since the challenge test data consists of microphone speech, it was downsampled from 16kHz to 8 kHz before MFCC features extraction.

3.2. Voice activity detection

In addition to energy-based VAD from Kaldi Toolkit and ASR based VAD [23] in this work we investigated our new neural network based VAD.

This work adapts the U-net [24] architecture to the task of speech activity detection. Such architecture was originally introduced in biomedical imaging for semantic segmentation in order to improve precision and localization of microscopic images. It builds upon the fully convolutional network and is similar to the deconvolutional network. In a deconvolutional network, a stack of convolutional layers – where each layer halves the size of the image but doubles the number of channels – encodes the image into a small and deep representation. That encoding is then decoded to the original size of the image by a stack of upsampling layers.

Our U-net based VAD is built on a modified and reduced version of the original architecture. Figure 1 schematically outlines the proposed version of neural network. It takes 8kHz 23-dimensional MFCC features as input. Our VAD solution works with a half overlapping 2.56sec sliding window and a 1.28sec overlap. It should be noted that each MFCC vector is extracted for 25ms frame every 20ms. This results in 128×23 input features size for the neural network.

The goal of the neural network is to predict the 128 dimensional speech activity mask for every 2.56sec speech segment. Thus the resolution of the proposed speech detector is equal to 20ms. The final decoder layer is a sigmoid activated global average pooling layer. Its output is used as the speech activity mask.

The U-net is trained on artificially augmented data with speech labels obtained from the oracle handmade segmentation or using oracle ASR based VAD processing of clean version of the data.

To train the network, we have used a combination of binary cross entropy loss function and dice loss [25]. The latter aims to maximize the dice coefficient between predicted binary segmentation set $p_i \in P$ and ground truth binary labels set $g_i \in G$:

$$\mathcal{D} = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}, \quad (3)$$

where the sums run over the N frames.

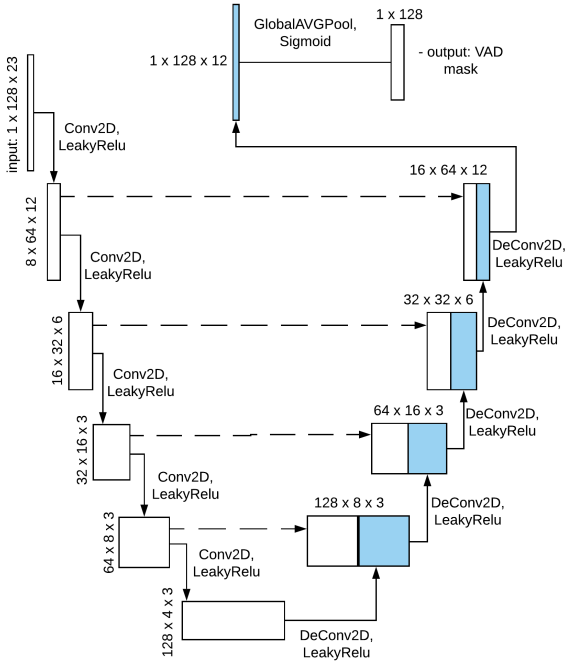

Figure 1: *U-net based VAD architecture*

3.3. Embedding extractors

We considered deep speaker embedding extractor with the most popular residual network architecture named ResNet34 and a deeper ResNet50 network [10].

Table 1 describes ResNet34 architecture we used. The key block of ResNet34 is ResNetBlock. It consists of two convolutional layers with 3×3 filters. ReLU activation follows each convolutional layer, and Maxout activation is used for embedding extraction. We apply batch normalization technique to stabilize and speed up network convergence. The settings for ResNet34 embedding extractors training were borrowed from [5].

Table 1: *Embedding extractor based on ResNet34 architecture configuration.*

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

More complex ResNet50 architecture contains three convo-

lutional layers in ResNetBlock with 1×1, 3×3, and 1×1 masks. Additionally, we used SE (Squeeze-and-Excitation) blocks [26] in each ResNetBlock.

3.4. Backend

In this work, we used Cosine Similarity (CS) and Cosine Similarity Metric Learning (CSML) for scoring. Additionally, adaptation and score normalization were applied.

3.4.1. CS and CSML

We used CS to distinguish *speaker embeddings*:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathbf{x_1}^T \mathbf{x_2}}{\|\mathbf{x_1}\|\|\mathbf{x_2}\|}, \quad (4)$$

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

As an alternative scoring model CSML approach was used for *speaker verification*. According to the original idea a linear transformation $\mathbf{A}$ was learned to compute cosine distance for a pair $(\mathbf{x_1}, \mathbf{x_2})$ as follows:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{A}) = \frac{(\mathbf{A x_1})^T (\mathbf{A x_2})}{\|\mathbf{A x_1}\|\|\mathbf{A x_2}\|}, \quad (5)$$

where the transformation matrix $\mathbf{A}$ is upper triangular. However, unlike [27] the triplet loss objective function was used for $\mathbf{A}$ training. The metric learning was performed similar to the way it was done in [28] using TensorFlow framework.

3.4.2. Domain adaption

In this work, we used simple domain adaptation procedure [29] based on centering on in-domain set (mean speaker embedding subtraction). The mean vector is calculated using adaptation set in this case.

3.4.3. Score normalization

Additionally, scoring systems normalization technique from [30] was used. For a pair $(\mathbf{x_1}, \mathbf{x_2})$ the normalized score can be estimated as follows:

$$\hat{\mathcal{S}}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) - \mu_1}{\sigma_1} + \frac{\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) - \mu_2}{\sigma_2}, \quad (6)$$

where the mean μ_1 and standard deviation σ_1 are calculated by matching $\mathbf{x_1}$ against impostor cohort and similarly for μ_2 and σ_2. A set of the *n* best scoring impostors were selected for each embedding pair when means and standard deviations are calculated.

4. Implementation details

Here we describe speaker recognition systems and datasets used for their training.

4.1. Datasets

In our experiment, we used three groups of training data:

- **TrainData-I** includes VoxCeleb1 [31] (without test data), VoxCeleb2 [32] and SITW [33] and their augmented versions. Augmentation was partially performed using standard Kaldi augmentation recipe (babble, music and noise) using the freely available MUSAN datasets[1]. Reverberation was performed using the impulse response generator based on [34]. Four different RIRs were generated for each of 40,000 rooms with a varying position of sources and destructors. It should be noted that, in contrast to the original Kaldi augmentation, we reverberated both speech and noise signals. In this case different RIRs generated for one room were used for speech and noise signals respectively. Thus we obtained more realistic data augmentation. We have already used this approach in our previous studies [15]. Energy-based VAD from Kaldi Toolkit was used to preprocess all samples from the database. The final database consists of approximately 5,200,000 samples (7,562 speakers);
- **TrainData-II** contains VoxCeleb1Cat (without test data) and VoxCeleb2Cat (without test data) and their augmented versions. We concatenated all segments from the same session into one file.Augmented data was generated using standard Kaldi augmentation recipe (reverberation, babble, music and noise) using the freely available MUSAN and RIR datasets[1]. Energy-based VAD from Kaldi Toolkit was used to preprocess all samples from the database. The final database consists of approximately 830,000 samples (7,146 speakers);
- **TrainData-III** is similar to TrainData-I, but ASR based VAD [23] was used to preprocess the examples from the database instead of the energy-based VAD;
- **TrainData-IV** is similar to TrainData-II, but it contains only VoxCeleb2Cat (without test data) and its augmented version. The final database consists of approximately 727,800 samples (5,994 speakers).

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

4.2. Extractors

ResNet34-MFB80-AM-TrainData-I: This system is based on ResNet34 embedding extractor. The key feature of this extractor is the use of high dimensional input features (80 dimensional MFB), as well as Maxout activation function on the embedding layer. Local CMN- and global CMVN-normalization are used to normalize extracted MFB features. This extractor was trained on short segments with the fixed 2 sec length and using AM-Softmax loss. Parameters m and s were respectively equal to 0.2 and 30 during the whole training stage. The learning rate was equal to 0.001 on the first two epochs, then it was decreased by a factor of 10 for each next epoch. TrainData-I was used for training.

Xvect-FTDNN-TrainData-I: This system is based on the factorized TDNN embedding extractor [1]. The main idea is that TDNN pre-pooling layers of the x-vector system are replaced by factorized TDNN with skip connections. Factorization of the weight matrix into two low-rank matrices, with one of them constrained to be semi-orthogonal, helps to reduce the number of neural network parameters. Using skip connections allows to solve the problem of gradient vanishing and makes training process more stable. In our speaker embedding extractor, described in more details in [15], we slightly modified the original skip connections and reduced the size of TDNN layers.

ResNet34-MFB80-D-TrainData-I: This extractor is similar to ResNet34-MFB80-AM-TrainData-I, with the difference of using a fine-tuning procedure by means of D-Softmax loss function.

ResNet50-SE-MFB80-AM-TrainData-I: This system is based on ResNet50 embedding extractor with SE blocks. Input features, training procedure and etc. were equivalent to

ResNet34-MFB80-AM-TrainData-I system.

ResNet34-MFCC40-AM-TrainData-I: This extractor is similar to ResNet34-MFB80-AM-TrainData-I, but uses 40 dimensional MFCC features as input. Local CMN- and global CMVN-normalization are applied.

ResNet34-MFB80-AM-TrainData-II (2s): This extractor is similar to ResNet34-MFB80-AM-TrainData-I, but was trained using TrainData-II dataset. AM-Softmax loss was used for the training with parameter s equal to 30 during the whole training stage and parameter m equal to 0.001 for the first epoch and to 0.2 for the next epochs. The initial value of learning rate was set to 0.001. The learning rate was decreased by a factor of 10 every next epoch.

ResNet34-MFB80-AM-TrainData-II (1s):. This extractor is similar to ResNet34-MFB80-AM-TrainData-II, but it was trained using only 1 sec duration chunks. Each 100 MFB speech frames with no overlap extracted from all samples of the TrainData-II were used for training. AM-Softmax loss was used for training, parameter s was equal to 30 during the whole training stage, parameter m was set to 0.2 for all epochs. The learning rate was the same as in ResNet34-MFB80-AM-TrainData-II (2s) system.

Xvect-Ext-TDNN-LSTM-TrainData-III: This extractor is described in [15]. The system is the extended version [7] of the original x-vector extractor, but with 9th layer replaced by LSTM-layer with cell dimension of 512, delay in the recurrent connections equal to -3, and both recurrent and non-recurrent projection dimension equal to 256. The LSTM layer context was reduced to 3. This embedding extractor was trained on TrainData-III.

ResNet34-MFB80-AM-TrainData-IV (1s): This extractor is similar to ResNet34-MFB80-AM-TrainData-II, but it was trained on 1 sec speech chunks obtained from TrainData-IV dataset in the same way as it was done for ResNet34-MFB80-AM-TrainData-II.

5. Experiments

5.1. Experimental setup

All experiments described further in this paper were performed with the use of VoxCeleb1 [35] and VOiCES 2019 challenge [2, 3] datasets. The results are presented in terms of EER and minDCF for $P_{tar} = 10^{-2}$ performance metrics.

5.2. Preliminary investigation

Our first goal was to investigate SR systems performance for the original full length testing protocols. Thus, table 2 demonstrates the experimental results obtained for the original "VOiCES dev", "VOiCES eval" and "VoxCeleb1-O cleaned" protocols.

We experimented with factorized TDNN based x-vector, ResNet34 and more complex ResNet50-SE networks. It should be noted that we used VOiCES eval part as a development (adaptation) set for the VOiCES dev part and vice versa. For the VoxCeleb1-O cleaned set we used a subset of 200k randomly selected clean files from VoxCeleb1 and VoxCeleb2 datasets in order to perform system adaptation and score normalization procedures. Top 10% of the impostor scores were used to perform s-normalization. In these experiments we focused on different VAD and backend model configurations. We also experimented with the promising D-Softmax based loss function to improve system performance in case of ResNet34-MFB80-D-TrainData-I.

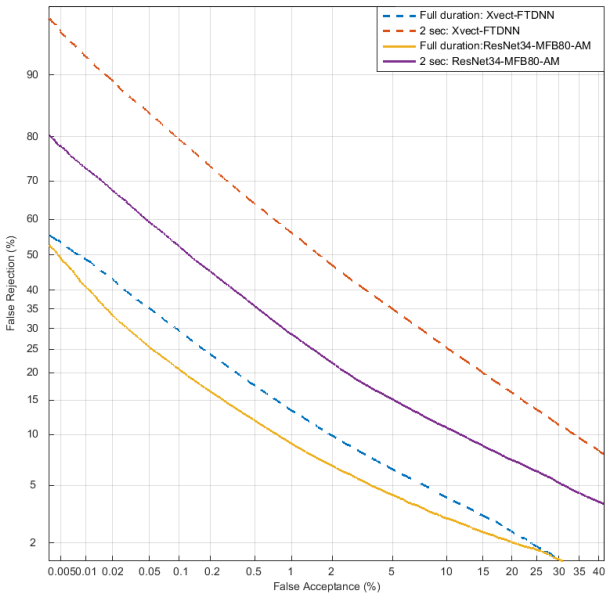

Figure 2: *DET curves of Xvec-FTDNN-TrainData-I and ResNet34-MFB80-AM-TrainData-I embedding extractors for full and 2 sec duration of test waveforms from VOiCES (eval set).*

5.3. Short utterance speaker recognition

In order to compare our SR systems performance for short utterances with those presented in [12, 13] the special dataset was generated from the VoxCeleb1 corpus according to the description from [12]: only files longer than 6 seconds (87010 utterances) were selected. A comparison protocol was generated by randomly sampling 100 target and 100 imposter pairs for each speaker from a total of 1,251 speakers in VoxCeleb1, resulting in 250,048 unique comparisons. We didn't succeed in obtaining the same size protocol as in [12] because not all speakers had the necessary 100 target comparisons with samples longer than 6 seconds. But since the original model proposed in [12] is freely available, we compared it with our ResNet34-MFB80-AM-TrainData-IV (1s) proposed above using the generated protocol. The comparison results of these models are shown in Table 3. The experiments were carried out in the same way as in [12, 13] without using any VAD.

During our experiments on short duration utterances, we used the following settings:

- as enrollment samples, we used only full duration original files;
- as test samples, we used only the first 1, 2 and 5 seconds of speech in each file. If speech duration was less than required, we used all available speech in this file and didn't change the protocol. We applied CMN with a 3-second sliding window on these segments in the following way: using VAD segmentation we accumulated the required amount of features, and applied CMN only to them discarding information redundant for the normalization;
- fixed protocol for all durations.

Table 3 demonstrates the comparison of all the described systems in case of different test samples duration (1 sec, 2 sec, 5 sec and full duration) for the same protocols in terms of EER.

Table 2: *Results of investigated systems for VOiCES (eval set), VOiCES (dev set) and VoxCeleb1-O (cleaned) protocols.*

Embedding extractor	Settings	VOiCES (eval set)		VOiCES (dev set)		VoxCeleb1-O (cleaned)	
		minDCF	EER, %	minDCF	EER, %	minDCF	EER, %
ResNet34-MFB80-AM-TrainData-I	CS backend	0.366	5.31	0.155	1.30	0.193	1.96
	+ mean adapt.	0.327	4.88	**0.144**	1.32	0.195	1.89
	+ s-norm.	0.319	4.82	0.169	1.30	0.187	1.78
	+ U-net VAD	**0.300**	**4.52**	0.157	**1.12**	0.173	1.76
Xvect-FTDNN-TrainData-I	CSML backend	0.496	7.12	0.253	2.28	0.363	4.15
	+ mean adapt.	0.426	6.03	0.234	1.99	0.381	4.18
	+ s-norm.	0.408	5.74	0.230	1.84	0.357	3.98
	+ U-net VAD	0.390	5.81	0.242	2.00	0.366	4.48
ResNet34-MFB80-D-TrainData-I	CS backend	0.419	5.36	0.220	2.06	0.241	2.12
ResNet50-SE-MFB80-AM-TrainData-I	CS backend	0.415	5.75	0.179	1.56	0.222	2.07
ResNet34-MFCC40-AM-TrainData-I	CS backend	0.405	6.03	0.178	1.38	0.236	2.14
ResNet34-MFB80-AM-TrainData-II	CS backend	0.447	6.25	0.193	1.27	0.151	1.45
	+ mean adapt.	0.383	5.81	0.185	1.35	0.148	1.44
	+ s-norm.	0.366	5.78	0.201	1.42	0.152	**1.39**
	+ U-net VAD	0.354	5.50	0.197	1.35	**0.142**	1.46
Xvect-Ext-TDNN-LSTM-TrainData-III	CSML backend, ASR VAD, s-norm.	0.349	5.16	–	–	–	–

In order to simplify the experiments and improve the reproducibility of our results, we did not use normalization and U-net based VAD for these tests.

Additionally, Fig. 2 presents Detection Error Tradeof (DET) curves of Xvec-FTDNN-TrainData-I and ResNet34-MFB80-AM-TrainData-I embedding extractors for full and 2 sec duration samples from VOiCES (eval set).

6. Discussion

6.1. Result for full duration experiments

Having analyzed the results obtained in 5.2, we can make the following conclusions:

1. Systems based on ResNet architectures outperform x-vector based systems in all our experiments.
2. U-net based VAD helps to improve the quality of systems for difficult conditions compared to the standard Kaldi energy-based VAD.
3. S-normalization improves the performance of both extractor types on the majority of the test settings for VoxCeleb and VOiCES in terms of the reduction of the reference metric for each of the contests (EER/minDCF). ResNet34-MFB80-AM-TrainData-II/TrainData-I models tested on the VOiCeS dev set setting are the only cases where mean adaptation outperforms s-normalization.
4. Appropriate training data preparation is an important step. Comparison of ResNet34-MFB80-AM-TrainData-II and ResNet34-MFB80-AM-TrainData-I shows that despite the fact that both systems show good quality in various experiments, the more task-oriented training data preparation can significantly improve the quality of systems.
5. Increasing the dimension of acoustic features helps to improve the quality of the systems. This statement is based on the results obtained for ResNet34-MFCC40-AM-TrainData-I and ResNet34-MFB80-AM-TrainData-I extractors which use relatively close features but not the same.

From the results presented in Table 2 one can see that the best performing system for VOiCES protocols is ResNet34-MFB80-AM-TrainData-I. It outperforms our previous best single system (Xvect-Ext-TDNN-LSTM-TrainData-I) submitted to the VOiCES challenge [15].

The obtained results allow to conclude that D-Softmax based loss training does not help to improve ResNet34-MFB80 performance. We also did not achieve any improvement by using more complex ResNet50-SE based extractor in comparison with ResNet34. We suppose that it is caused by the more complex model being overfitted in this case.

For the VoxCeleb1-O (cleaned) protocol ResNet34-MFB80-AM-TrainData-II (2s) is the top performing system. It was trained on 2 sec speech chunks of TrainData-II dataset.

6.2. Result for the experiments with short utterances

Taking into account the results from Table 3 obtained for modified Voxceleb1 dataset we can summarize that, while the protocol generation procedure was as close as possible to that described in [12], our results were somewhat different but comparable to the ones published in [12, 13]. The observed differences can be attributed to the differences in test protocols. Nevertheless, synchronous testing showed significantly better quality of the proposed model for various short durations (1 sec, 2 sec and 5 sec). It should be noted that the model proposed in [13] also demonstrated significantly better quality on short pronunciations compared to the model [12]. Unfortunately, we were unable to test the [13] model due to its public unavailability.

Having analyzed the results obtained in 5.3, we can make the following conclusions:

1. Training ResNet based embedding extractors on short utterances leads to an improvement in its performance for shorter durations at a cost of performance degradation on full durations. This is confirmed by the results of ResNet34-MFB80-AM-TrainData-I model, trained on 1-second and 2-second segments. Thus, it is possible to

Table 3: *The results of the publicly available model from [12] and the proposed ResNet34-MFB80-AM-TrainData-IV on the short utterances protocol (generated from VoxCeleb1).*

Embedding extractor	EER, % (1s)	EER, % (2s)	EER, % (5s)
Thin ResNet34 [12]	12.71	6.59	3.34
ResNet34-MFB80-AM-TrainData-IV (1s)	9.91	4.48	2.26

Table 4: *Results of the investigated systems for VOiCES (eval set), VOiCES (dev set) and VoxCeleb1-O (cleaned) protocols in relation to different length of test waveform.*

Embedding extractor	VOiCES (eval set) EER, % (1s/2s/5s/full)	VOiCES (dev set) EER, % (1s/2s/5s/full)	VoxCeleb1-O (cleaned) EER, % (1s/2s/5s/full)
ResNet34-MFB80-AM-TrainData-II (2s), CS backend	16.96/10.14/7.77/6.25	8.70/3.67/1.83/1.27	6.77/2.74/1.59/1.45
ResNet34-MFB80-AM-TrainData-II (1s), CS backend	19.85/14.27/12.61/11.17	9.70/5.07/3.23/2.63	6.87/3.18/2.11/1.99
ResNet34-MFB80-AM-TrainData-IV (1s), CS backend	20.17/14.76/13.24/10.80	10.13/5.15/3.27/2.46	7.13/3.54/2.23/2.09
ResNet34-MFB80-AM-TrainData-I (2s), CS backend	16.26/**9.46**/6.95/5.31	8.97/4.15/1.97/1.30	8.04/3.47/2.08/1.96
ResNet34-MFB80-AM-TrainData-I (1s), CS backend	**16.21**/10.63/9.20/8.00	8.43/4.35/2.57/2.06	7.51/4.03/2.60/2.57
Xvect-FTDNN-TrainData-I (2–3s), CSML backend	25.62/15.97/11.22/7.12	20.88/10.82/5.08/2.28	19.45/9.96/4.80/4.15
ResNet50-SE-MFB80-AM-TrainData-I (2s), CS backend	17.67/9.89/7.12/5.75	10.22/4.12/2.17/1.56	8.86/3.73/2.18/2.07

slightly improve the quality of the systems for short durations in this way.

2. TDNN based x-vector systems degrade more than ResNet systems on short segments test. We can observe this effect by analysing the results for Xvect-FTDNN-TrainData-I and ResNet34-MFB80-AM-TrainData-II (1s) systems on the VOiCES (dev set) and Xvect-FTDNN-TrainData-I and ResNet34-MFB80-AM-TrainData-II (2s) on VOiCES (eval set).

3. The better the acoustic conditions of the dataset are the more visible relative performance degradation can be achieved on it. This is confirmed by the results obtained for described x-vector and ResNet systems on VoxCeleb1-O cleaned and VOiCES eval datasets.

7. Conclusion

Obtained results confirm that ResNet architectures allow to improve the quality of speaker verification for both long-duration and short-duration utterances, in comparison with the standard x-vector approach. Our best performing system for VOiCES protocols is ResNet34 based system built on 80 dimensional MFB features. It outperforms our previous best single system submitted to the VOiCES challenge. It is important to note that the appropriate training data preparation can significantly improve the quality of the final SR systems. We also should note that utilization of the proposed U-net based VAD (instead of energy based VAD), scoring model, mean adaptation and score normalization techniques provides additional performance gains for SR systems.

8. Acknowledgements

This work was partially financially supported by the Government of the Russian Federation (Grant 08-08) and by the Foundation NTI (contract 20/18gr) ID 0000000007418QR20002.

9. References

[1] Jesús Villalba et al., "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and speakers in the wild evaluations," *Computer Speech & Language*, vol. 60, pp. 101026, 2020.

[2] Colleen Richey et al., "Voices obscured in complex environmental settings (VOiCES) corpus," in *INTERSPEECH 2018*, Hyderabad, India, September 2018, pp. 1566–1570.

[3] Mahesh K. Nandwana, Julien van Hout, Mitchell McLaren, Colleen Richey, Aaron Lawson, and Maria A. Barrios, "The VOiCES from a distance challenge 2019 evaluation plan," in *arXiv preprint arXiv: 1902.10828 [eess.AS]*, March 2014.

[4] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH 2017*, Stockholm, Sweden, August 2017, pp. 999–1003.

[5] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," in *arXiv:1910.12592 [eess.AS]*, October 2019.

[6] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey 2010*, Brno, Czech Republic, June–July 2010, p. 14.

[7] David Snyder et al., "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019*, Brighton, UK, May 2019, pp. 5796–5800.

[8] Sergey Novoselov, Andrey Shulipa, Ivan Kremnev, Alexander Kozlov, and Vadim Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," in *Odyssey 2018*, Les Sables d'Olonne, France, June 2018, pp. 378–385.

[9] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "SphereFace: Deep hypersphere embedding for face recognition," in *CVPR 2017*, Honolulu, Hawaii, USA, July 2017, pp. 6738–6746.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR 2016*, Las Vegas, Nevada, USA, June 2016, pp. 770–778.

[11] Galina Lavrentyeva et all., "Audio replay attack detection with deep learning frameworks," in *INTERSPEECH 2017*, Stockholm, Sweden, August 2017, pp. 82–86.

[12] Weidi Xie, Arsha Nagrani, Joon S. Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019*, Brighton, UK, May 2019, pp. 5791–5795.

[13] Amirhossein Hajavi and Ali Etemad, "A deep neural network for short-segment speaker recognition," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2878–2882.

[14] Weicheng Cai, Zexin Cai, Xiang Zhang, Xiaoqi Wang, and Ming Li, "A novel learnable dictionary encoding layer for end-to-end language identification," in *ICASSP 2018*, Calgary, Canada, April 2018, pp. 5189–5193.

[15] Sergey Novoselov, Alexey Gusev, Artem Ivanov, Timur Pekhovsky, Andrey Shulipa, Galina Lavrentyeva, Vladimir Volokhov, and Alexander Kozlov, "STC speaker recognition systems for the VOiCES from a distance challenge," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2443–2447.

[16] Danwei Cai, Xiaoyi Qin, Weicheng Cai, and Ming Li, "The DKU system for the speaker recognition task of the 2019 VOiCES from a distance challenge," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2493–2497.

[17] Pavel Matejka et all., "Analysis of BUT submission in far-field scenarios of VOiCES 2019 challenge," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2448–2452.

[18] David Snyder, Jesús Villalba, Nanxin Chen, Daniel Povey, Gregory Sell, Najim Dehak, and Sanjeev Khudanpur, "The JHU speaker recognition system for the VOiCES 2019 challenge," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2468–2472.

[19] Tomohiro Nakatani, Takuya Yoshioka, Keisuke Kinoshita, Masato Miyoshi, and Biing-Hwang Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 1717–1731, 2010.

[20] Feng Wang, Weiyang Liu, Haijun Liu, and Jian Cheng, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[21] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *CVPR 2019*, Long Beach, California, USA, June 2019, pp. 4690–4699.

[22] Lanqing He, Zhongdao Wang, Yali Li, and Shengjin Wang, "Softmax dissection: Towards understanding intra- and inter-class objective for embedding learning," in *arXiv:1908.01281 [cs.CV]*, August 2019.

[23] Ivan Medennikov et al., "The STC ASR system for the VOiCES from a distance challenge 2019," in *INTERSPEECH 2019*, Graz, Austria, September 2019, pp. 2453–2457.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI 2015*, Munich, Germany, October 2015, pp. 234–241.

[25] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV 2016*, Stanford, CA, USA, October 2016, pp. 565–571.

[26] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *CVPR 2018*, Salt Lake City, Utah, USA, June 2018, pp. 7132–7141.

[27] Hieu V. Nguyen and Li Bai, "Cosine similarity metric learning for face verification," in *ACCV 2010*, Queenstown, New Zealand, November 2010, pp. 709–720.

[28] Sergey Novoselov, Vadim Shchemelinin, Andrey Shulipa, Alexander Kozlov, and Ivan Kremnev, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition," in *INTERSPEECH 2018*, Hyderabad, India, September 2018, pp. 2242–2246.

[29] Jahangir M. Alam, Gautam Bhattacharya, and Patrick Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Odyssey 2018*, Les Sables d'Olonne, France, June 2018, pp. 176–180.

[30] Daniel Colibro, Claudio Vair, Emanuele Dalmasso, Kevin Farrell, Gennady Karvitsky, Sandro Cumani, and Pietro Laface, "Nuance–Politecnico di Torino's 2016 NIST speaker recognition evaluation system," in *INTERSPEECH 2017*, Stockholm, Sweden, August 2017, pp. 1338–1342.

[31] Arsha Nagrani, Joon S. Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *INTERSPEECH 2017*, Stockholm, Sweden, August 2017, pp. 2616–2620.

[32] Joon S. Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep speaker recognition," in *INTERSPEECH 2018*, Hyderabad, India, September 2018, pp. 1086–1090.

[33] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The speakers in the wild (SITW) speaker recognition database," in *INTERSPEECH 2016*, San Francisco, USA, September 2016, pp. 818–822.

[34] Jont B. Allen and David A. Berkley, "Image method for efficiently simulating small-room acoustics," *Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[35] Arsha Nagrani, Joon S. Chung, Weidi Xie, and Andrew Zisserman, "VoxCeleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, pp. 101027, 2020.