Description of the DSP AGH systems for the SdSV Challenge

Magdalena Rybicka, Stanisław Kacprzak, Marcin Witkowski and Konrad Kowalczyk

Digital Signal Processing Group
Department of Electronics
AGH University of Science and Technology
Krakow, Poland

{mrybicka, skacprza, witkow, konrad.kowalczyk}@agh.edu.pl

Abstract

In the following we describe systems used to generate the DSP AGH submission to the Short-duration Speaker Verification Challenge 2020, in which we address the problem of speaker verification from utterances of short duration with cross-language domain mismatch between enroll and test conditions. We perform domain adaptation directly in speaker embedding space using consistent generative adversarial network (CycleGAN), and present a suitable network architecture and loss to operate on vector embeddings.

Index Terms: short-duration speaker verification, cycleGAN, domain adaptation, ResNet18, speaker embedding

1. Proposed system

This section provides an overview of the proposed system for speaker verification from short-duration speaker recordings with cross-language trials. Two major problems posed by the SdSV Challenge 2020 dataset include: (i) different utterance duration in enrollment (3 to 120 s) and test (up to 8 s) and (ii) mismatch in language of enroll and test utterances, i.e. enroll consists of only Farsi (Persian), whilst test set consists of English and Farsi, i.e. it contains unobserved data. The proposed system, depicted in Fig. 1, enables to overcome both of these problems. First, we develop a novel DNN architecture based on modifying the smaller version of the ResNet, namely ResNet18, which mainly concerned modifications to improve feature extraction from audio input and gathering statistics with Learnable Dictionary Encoding (LDE) [1, 2]. We then propose to perform domain adaptation using shallow CycleGAN applied directly in the speaker embedding space. Note that in contrast to approaches in [3, 4], domain adaptation in the embedding space allows to use a much simpler DNN architecture of generators and discriminators, which significantly reduces the computational cost. Furthermore, in contrast to [3, 4], embeddings from both source and target domains undergo domain adaptation and we provide adjustments in consistency and identity loss functions to operate on embeddings. Target-domain embeddings are mean centralized before they are fed to a classical target-domain-adapted backend for speaker verification.

2. Speaker embedding extraction

In this section, we describe modifications made to the original ResNet18 [5] architecture which facilitate audio feature extraction and propose further improvements to achieve superior performance in speaker verification for utterances of short duration.

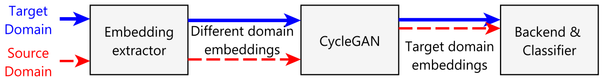


Figure 1: *Diagram of the proposed system with domain adaptation in embedding space using CycleGAN.*

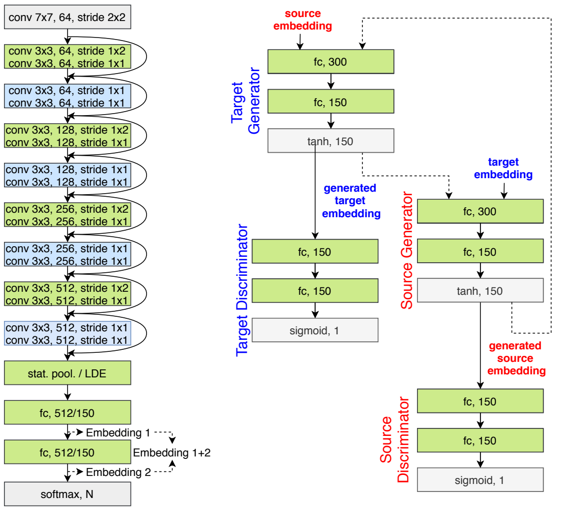
2.1. Modified ResNet architecture (mR18)

In this section we present the basic structure of our proposed ResNet-based architecture. In order to tackle speaker verification of short utterances, we adjust 18-layer variant of ResNet network architecture which we modify to enable speaker embedding extraction. The diagram of the proposed modified ResNet18-based network structure is depicted in Fig. 2(a). Input features are first processed with a 2D convolutional layer with filter kernel size of 7x7, downsampling stride of 2x2 and output of 64 channels. The output of the first layer is fed to the residual part of the architecture, which is composed of 4 segments, each containing 2-layer blocks. In ResNet18 2D convolutional layer with kernel size of 3x3 is the basic unit. The number of channels per segment respectively are {64, 128, 256, 512}. In each segment the first layer downsamples input along the frequency axis with stride of size 2, which allows the neural network to focus more on the temporal dependency between the frames. The residual part is followed by the statistics pooling layer and two fully-connected layers with size of 512, and a softmax layer with the number of outputs equal to the number of classes in the trainig dataset. We will refer to this architecture as a modified ResNet18 (mR18).

2.2. Improvements to mR18 and embedding selection

To further enhance neural network ability of generating robust embeddings for short input sentences, we introduce the following improvements to the mR18 network. (i) We replace standard Rectified Linear Unit (ReLU) activation functions commonly used e.g. in TDNNs and original ResNets, with the so-called Leaky ReLU functions [6] with the aim to avoid zero gradient occurrence for negative function arguments (with parameter set to 0.2). (ii) We propose to replace the traditional statistics pooling layer with the Learnable Dictionary Encoding (LDE) [1, 2] layer, which contains dictionary of component centers, learned during the network training. The LDE weights are estimated for each time frame and component center, and they are subsequently used to obtain the encoding of the entire utterance with

This research received financial support from the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union under the European Regional Development Fund and was supported in part by PLGrid Infrastructure.



(a) *mR18 / imR18*

(b) *Shallow CycleGan*

Figure 2: *The proposed architectures of (a) ResNet18-based DNNs, and (b) the shallow CycleGAN.*

respect to each class center. The final vector is formed as a concatenation of the obtained single component encodings. In our architecture there are 64 components. (iii) We replace the standard loss function with a margin-based softmax cross-entropy loss function which incorporates an additive angular margin in the angular function. This so-called Additive Angular Softmax (AAS) [7, 8] enforces better separation between the class representations during the neural network training. The margin parameter is set at 0.3 and scale at 30. Architecture with so far described improvements will be referred as improved mR18 (imR18). (iv) Since shorter utterances contain less phonetic information, we reduce the size of the two fully-connected layers from 512 to 150. As suggested in [9], lowering the size of these layers, and thereby also of speaker embeddings, may support generating more discriminative speaker representation.

3. Domain adaptation in embedding space using CycleGAN

Inspired by the use of CycleGAN [10] for domain adaptation in speaker recognition [3, 4], we propose to perform domain adaptation, however, in an embedding space. Operating on the already extracted embedding vectors, as opposed to the time-frequency domain representation of the audio signal, is beneficial as it enables a significant reduction of the network complexity of the CycleGAN model and avoids re-computation of embeddings after domain adaptation. Since we perform domain adaptation in the embedding space, we expect to achieve vector translation to the region of the embeding space which corresponds to the *target* domain. This non-linear transformation can be treated as an alternative to a linear LN-WCCN [11].

3.1. CycleGAN architecture

Operating in the embedding space significantly reduces the size of the input to the model (a single vector instead of a 2D time-frequency representation). For that reason we experiment mainly with shallow architectures composed of a couple of fully connected layers with *tanh* activation function. The CycleGAN consists of two generators and two discriminators, the architectures of which are depicted in Fig. 2. Generators are responsible for the translation between the two domains and discriminators make binary decisions if the input vector belongs or does not belong to the domain. Both networks operate on length normalized embeddings and the generator normalizes its output as a post processing step. Note that although we are interested only in a one-way mapping, CycleGAN learns the mapping in both directions to allow regularization in the form of cycle-consistency, which requires reconstruction of the original features with minimum reconstruction error (by transferring them back to the original domain).

3.2. CycleGAN loss function

Formulation of the proposed CycleGAN model in embedding space follows the model presented in [10, 3]. The generator $G_{S\to T}$ is trained to learn mapping from the *source* domain S to the *target* domain T. The training data X_S and X_T consists of elements drawn from two separate distributions $x_S \sim p_S(x)$ and $x_T \sim p_T(x)$. The discriminator D_T is trained to recognize elements from domain T. As in [3], the GAN loss function is defined in terms of the mean square error defined as

$$L_{GAN_{S\to T}}=\mathbb{E}_{x\sim p_S}[D_T(G_{S\to T}(x))^2]+\mathbb{E}_{x\sim p_T}[(D_T(x)-1)^2] \quad (1)$$

We define the loss function $L_{GAN_{T\to S}}$ for $G_{T\to S}$ and D_S similarly. The CycleGAN is regularized by additional losses. Since our model operates on embeddings, we apply the cosine distance d_{cos} as a similarity measure. The cycle consistency loss and the identity loss are defined, respectively, as

$$\begin{aligned} L_{cyc} = &\mathbb{E}_{x\sim p_S}[d_{cos}(G_{T\to S}(G_{S\to T}(x)),x)] + \\ &\mathbb{E}_{x\sim p_T}[d_{cos}(G_{S\to T}(G_{T\to S}(x)),x)]\,, \end{aligned} \quad (2)$$

$$\begin{aligned} L_{id} = &\mathbb{E}_{x\sim p_S}[d_{cos}(G_{T\to S}(x),x)] + \\ &\mathbb{E}_{x\sim p_T}[d_{cos}(G_{S\to T}(x),x)]\,, \end{aligned} \quad (3)$$

with the latter loss function enforcing transformation invariance of the embeddings from the generators output domains. Finally, the total loss for CycleGAN is given by

$$L_{Total} = L_{GAN_{S\to T}} + L_{GAN_{T\to S}} + \lambda_{cyc}L_{cyc} + \lambda_{id}L_{id}\,, \quad (4)$$

where λ_{cyc} and λ_{id} are weigthing coefficients.

4. Target-domain speaker verification

For system backend, we apply mean centralization, LDA dimension reduction with vector length normalization, PLDA classification and score normalization using adaptive s-normalisation [12]. Although standard processing is used in our system backend, we focus on evaluating the influence of selecting embeddings, with and without domain adaptation, from different available datasets to appropriately adapt system backend for enroll and test. In particular, we show that data selection for mean centralization largely affects the performance.

5. Datasets, system training and setup

In this section, we provide the descriptions of datasets, system training procedure, and system setup. Core datasets used in the development of the systems are VoxCeleb1 [13] and VoxCeleb2 [14] along with training subset provided in the SdSV Challenge for Task 2 (SdSV-train). VoxCeleb datasets contain 1 276 888 utterances from 7 323 speakers, while SdSV-train contains 85 764 recordings from 588 speakers. We apply 4 types of augmentations: reverberation (RIRs from small and medium sized rooms), babble noise with speech with 3-7 overlapping speakers, music, and noise from MUSAN [15] corpus. As input, we use 64-band Mel-filter Bank coefficients with frames of 25 ms duration and 10 ms overlap. The energy-based VAD with energy threshold set to 3.5 is used. In early developments, the size of fully connected layers is set to a typical value of 512, followed by LDA reduction to 200. In latter experiments, the size of fully connected is reduced to 150 with LDA reduction to 125. In all experiments, the s-norm cohort contains 10 000 utterances from SdSV-train with subset of 10% of top-scoring. System structure including frontend and backend processing are based on the Kaldi pipeline, whereas NNs are implemented in TensorFlow [8, 16].

In experiments with CycleGAN, the GAN-transformed vectors are also used for backend training. CycleGAN training requires two datasets. The target domain dataset is generated using embedings extracted from the entire SdSV-train, i.e. from recordings with spoken Farsi language. The source domain dataset is composed of embeddings extracted from the subset of English samples from VoxCeleb1 and VoxCeleb2. Both CycleGAN training sets contain the same number of embeddings. Note that by this setup, duration mismatch between source and target domains is additionally enforced. Since we observed that target embeddings are not completely invariant to the CycleGAN transformation, in the experiments we transform both sets to the new target domain. CycleGAN is implemented using PyTorch [17]. During training, batches of size of 32 are randomly sampled from each domain, Adam optimizer with momentum $\beta_1 = 0.5$, $\beta_2 = 0.999$ is used, while other parameters follow [10], namely: $\lambda_{cyc} = 10.0$ and $\lambda_{iden} = 5.0$. Final model is trained for 850 epochs.

As evaluation measures we use metrics specified in the evaluation plan [18], namely the Equal Error Rate (EER) and minimal Detection Cost Function (minDCF) with parameters set to $C_{Miss} = 10$, $C_{FalseAlarm} = 1$ and $P_{Target} = 1$.

6. Single system development summary

In this section, we present the development of a single, proposed system based on improvements made to initial mR18 DNN-based system. We focused on ResNet-based architectures since they in our primary tests significantly outperformed TDNN-based systems on the SdSVC dataset.

The summary of key milestones in performance gain of speaker verification from short-duration utterances achieved by the system proposed in this report is presented in Fig. 3. All results presented in this section are obtained from the SdSVC online CodaLab submission system, using 70% of SdSVC evaluation data.

Having selected the network architecture (mR18), we increase accuracy by improvements made to the network architecture and training loss introduced as described in Sec. 2.2. The first two systems, denoted as *arch* and *impr arch*, are trained using both VoxCeleb datasets, while SdSV-train is used for backend adaptation. Slight improvement can be observed with the SdSV-train added to the speaker embedding extractor training, which is denoted as *train* in Fig. 3. In the next step, a much more significant gain in performance is achieved by proper selection of the short embedding size (namely reducing the size from 500 to 150) and performing domain adaptation with the proposed CycleGAN. Finally, we have experimented with using different data for backend adaptation, in particular we noticed large differences in performance in mean-centering for enroll and test sets depending on the data used in mean calculation. Choosing appropriate domains for performing *backend* adaptation with mean centering, brings about final improvements in the obtained single system results. Specifically, the highest gain has been achieved with mean computed using 85 764 Farsi embeddings for enrollment and combination of Farsi and English (171 528 embeddings) for test, which matches best the structure of evaluation trials of the SdSV challenge. In this experiment we apply data augmentation to subset of 1 milion utterances from VoxCeleb dataset and entire SDSV-train for LDA and PLDA training.

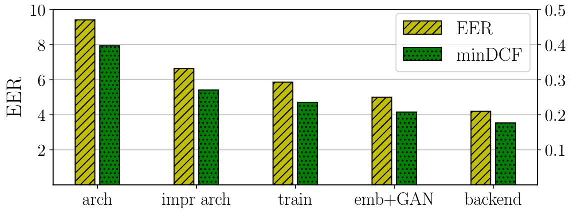


Figure 3: *Performance gain achieved by the proposed system.*

7. Final results in SdSV Challenge

In this section, we present the final results obtained by the proposed system with and without the application of the CycleGAN, and the results obtained by the fusion of several systems submitted in the challenge. The weights for linear fusion of the system scores were trained using the entire SdSV-train set using the Bosaris Toolkit [19].

Table 1 reports the EER and minDCF results obtained for the single system and a group of fused systems. For system fusion, we use 4 systems which achieved high performance on the leaderboard. The submitted fused scores are obtained using the following architectures:

1. imR18 DNN architecture with CycleGAN (we use system without embedding size reduction);
2. imR18 DNN architecture with embedding of length 150 extracted after the 1st fully connected layer;
3. imR18 architecture with data added for network training (i.e. Librispeech [20] dataset which contains 292 367 utterances of 5 831 speakers is added to the training data);
4. mR18 DNN architecture with embedding of length 150 extracted after the 2nd fully connected layer, with LDA dimensionality reduction to size of 75 (note that other systems use LDA dimensionality reduction to 125).

Table 1 presents the results of our final submissions to the SdSV Challenge 2020 for the progress and evaluation sets. We compare the results for the baseline system, which are taken from the leaderboard, with the proposed system containing the described improvements to the modified ResNet18 architecture. The single system is based on imR18 network with fully connected layer size of 150 and appropriate backend adaptation. We present two results which are obtained by the described processing with and without the application of the CycleGAN. All systems in final submission followed the same backend scenario as in the last experiment described in Sec. 6.

For the Challenge submission, all fused systems are based on the proposed modified ResNet18 architecture (either mR18 or imR18), trained using both VoxCeleb datasets and the SdSV-train, with mean centering strategy that matches evaluation trials (as in system *backend* in Fig. 3) and one of the systems used in fusion involves GAN domain adaptation.

As can be observed, in general the proposed system achieves very significant improvement in performance over the baseline system based on the ETDNN structure. Furthermore, the application of the proposed CycleGAN further improves the results. Finally, system fusion enables to achieve substantial improvement in performance over both considered single systems.

Table 1: *Final submission results (EER[%] and minDCF).*

System	Progress set	Evaluation set
Baseline (leaderboard)	10.67 / 0.432	10.67 / 0.432
Single system w/o GAN	4.37 / 0.183	4.36 / 0.182
Single system with GAN	4.23 / 0.177	4.21 / 0.177
Fusion	3.67 / 0.158	3.68 / 0.157

8. Conclusions

This report presents the outcome of our submission to the SdSV 2020 Challenge, in which we propose a system consisting of ResNet-based speaker embedding and domain adaptation using CycleGAN in speaker embedding space. The results of performed experiments demonstrate that an improved ResNet18-based architecture with the proposed domain adaptation support cross-language system adaptation with high-accuracy SdSV.

9. References

[1] W. Cai, J. Chen, and M. Li, “Exploring the encoding layer and loss function in end-to-end speaker and language recognition system,” in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[2] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, “A novel learnable dictionary encoding layer for end-to-end language identification,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5189–5193, 2018.

[3] P. S. Nidadavolu, J. Villalba, and N. Dehak, “Cycle-GANs for domain adaptation of acoustic features for speaker recognition,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 6206–6210.

[4] P. S. Nidadavolu, S. Kataria, J. Villalba, and N. Dehak, “Low-resource domain adaptation for speaker recognition using Cycle-GANs,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 710–717.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.90

[6] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[7] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Y. Liu, L. He, and J. Liu, “Large Margin Softmax Loss for Speaker Verification,” *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/interspeech.2019-2357

[9] A. Kanagasundaram, S. Sridharan, G. Sriram, S. Prachi, and C. Fookes, “A study of x-vector based speaker recognition on short utterances,” in *Proc. Interspeech 2019*, 2019, pp. 2943–2947.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[11] M. McLaren, M. I. Mandasari, and D. A. van Leeuwen, “Source normalization for language-independent speaker recognition using i-vectors,” *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pp. 55–61, 2012.

[12] O. Plchot, S. Matsoukas, P. Matejka, N. Dehak, J. Z. Ma, S. Cumani, O. Glembek, H. Hermansky, S. H. R. Mallidi, N. Mesgarani *et al.*, “Developing a speaker identification system for the darpa rats project.” in *ICASSP*, 2013, pp. 6768–6772.

[13] A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[14] J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep speaker recognition,” in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[15] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.

[16] H. Zeinali, L. Burget, J. Rohdin, T. Stafylakis, and J. H. Cernocky, “How to Improve Your Speaker Embeddings Extractor in Generic Toolkits,” *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6141–6145, 2018.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[18] K. A. Zeinali, Hossein nad Lee, J. Alam, and L. Burget, “Short-duration speaker verification (SdSV) challenge 2020: the challenge evaluation plan.” arXiv preprint arXiv:1912.06311, Tech. Rep., 2020.

[19] N. Brümmer and E. De Villiers, “The bosaris toolkit: Theory, algorithms and code for surviving the new dcf,” *arXiv preprint arXiv:1304.2865*, 2013.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.