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

SdSVC Challenge 2021: Tips and Tricks to Boost the Short-duration Speaker Verification System Performance

Aleksei Gusev[1,2], *Alisa Vinogradova*[1,2], *Sergey Novoselov*[1,2], *Sergei Astapov*[1]

[1]ITMO University, St. Petersburg, Russia
[2]STC-Innovations Ltd., St. Petersburg, Russia
{gusev-a, gazizullina, novoselov, astapov}@speechpro.com

Abstract

This paper presents speaker recognition (SR) systems for the text-independent speaker verification under the cross-lingual (English vs Persian) task (task 2) of the Short-duration Speaker Verification Challenge (SdSVC) 2021.

We present the description of applied ResNet-like and ECAPA-TDNN-like topology design solutions as well as an analysis of multi-session scoring techniques benchmarked on the SdSVC challenge datasets. We overview various modifications of the basic ResNet-like architecture and training strategies, allowing us to obtain the improved quality of speaker verification. Also, we introduce the alpha query expansion-based technique (αQE) to the enrollment embeddings aggregation at test time, which results in a 0.042 minDCF improvement from 0.12 to 0.078 for the ECAPA-TDNN system compared to the embeddings mean. We also propose a trial-level distance-based non-parametric imposter/target detector (KrTC) used to filter out the worst enrollment samples at test time to further improve the performance of the system.

Index Terms: SdSVC, speaker recognition, deep neural network, domain adaptation, ECAPA-TDNN, alpha query expansion, k-reciprocal

1. Introduction

The extension of voice biometrics methods to a wide range of tasks, such as cross-channel verification, recognition with the short file duration as well as far-field microphone task has led to the development of methods applied to the task of speaker recognition and contributed to an increased interest in this area. So in 2020, a number of competitions (Second VoxCeleb Speaker Recognition Challenge [1], Far-Field Speaker Verification Challenge 2020 [2], Short-duration Speaker Verification Challenge 2020 [3]) were held, aiming at developing a text-independent automatic speaker verification.

As a result, a number of methods have been developed, which allow getting higher quality on the test protocols of the competition. In [4] authors discuss the possibility of using metric learning for speaker recognition using the angular prototypical loss function. In [5] proposed of Dual Path Network (DPN) [6] architecture for task of speaker recognition. In [7] authors successfully use Sub-Centers AAM-softmax loss (SC-AAM). SC-AAM allows filtering noisy utterances out from train dataset using several centroids for a class. SdSVC 2021 continues the chain of these competitions and aims to develop methods of speaker recognition on short-duration utterances under the condition of cross-lingual speaker verification (English vs. Persian).

During participation in the competition, we reviewed a number of architectural solutions that allowed us to improve the quality of speaker recognition in the comparison with basic ResNet architectures. We also investigated multi-enroll aggregation techniques which further improved the performance of our systems on test.

Contribution:

1. Propose modification ResNet architecture derived by Network Architecture Search (NAS) for the speaker recognition task.
2. Propose αQE-based enrollment aggregation technique.
3. Propose k-reciprocal-based imposter / target trial detector.

2. System components

2.1. Acoustic Features

Input features for all systems presented in this paper are 80-dimensional Log Mel-filter bank (MFB) energies extracted from 16kHz raw input signals. 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 [8]. The U-net-based VAD [9, 10] was used after the CMN-normalization procedure. The details on the VAD training procedure are described in [11].

2.2. Embedding Extractors

We used two kinds of neural network architectures to process acoustic features – ResNet-like and ECAPA-TDNN-based systems.

ResNet-like The use of ResNet-like architectures significantly improves the results of the speaker verification algorithms on a large number of different test protocols [12, 13]. However, according to the latest research in the field of image processing [14, 15], better quality can be achieved with modifications to the standard ResNet topology if compared with the basic models. In this work, we have adopted a part of architectural modifications to the ResNet backbone from [15] to the task of speaker verification. We get gains in quality on VoxCeleb1 test and SdSVC test datasets with our ResNet-NAS compared to baseline ResNet-101 with bottleneck blocks replaced by basic blocks. The differences between the architectures are presented in more detail in the Table 1.

We use the Adaptive Curriculum Learning loss function (see subsection 2.5) to train this type of models. Parameters m and s are respectively equal to 0.35 and 32 during the whole training stage.

ECAPA-TDNN-based Emphasized Channel Attention, Propagation and Aggregation in TDNN (ECAPA-TDNN), newly proposed in [16], is a modification of the standard

http://dx.doi.org/10.21437/Interspeech.2021-1737

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

layer name	ResNet-NAS	ResNet101-like
Input	80 MFB log-energy	
Conv2D-1	3×3, stride 1	
Block-1	$\begin{bmatrix} 3 \times 3, 40 \\ 3 \times 3, 40 \end{bmatrix} \times 3$, st 1	$\begin{bmatrix} 3 \times 3, 50 \\ 3 \times 3, 50 \end{bmatrix} \times 3$, st 1
Block-2	$\begin{bmatrix} 3 \times 3, 80 \\ 3 \times 3, 80 \end{bmatrix} \times 6$, st 2	$\begin{bmatrix} 3 \times 3, 100 \\ 3 \times 3, 100 \end{bmatrix} \times 4$, st 2
Block-3	$\begin{bmatrix} 3 \times 3, 240 \\ 3 \times 3, 240 \end{bmatrix} \times 18$, st 2	$\begin{bmatrix} 3 \times 3, 200 \\ 3 \times 3, 200 \end{bmatrix} \times 23$, st 2
Block-4	$\begin{bmatrix} 3 \times 3, 240 \\ 3 \times 3, 240 \end{bmatrix} \times 9$, st 2	$\begin{bmatrix} 3 \times 3, 400 \\ 3 \times 3, 400 \end{bmatrix} \times 3$, st 2
StatsPooling	mean and std, 20×240	mean and std, 20×400
Flatten	4800	8000
Dense1	$[4800 \times 512] \times 2$, $MaxOut$	$[8000 \times 512] \times 2$, $MaxOut$
Dense2	$[512 \times N_{spk}]$	

Time Delay Neural Network (TDNN) architecture, containing Squeeze-Excitation (SE) blocks and Res2Net modules at the frame level and attentive statistic pooling (ASP) [17] instead of the usual statistic pooling. We use our implementation of ECAPA-TDNN architecture with the following parameters: the number of SE-Res2Net Blocks is set to 4 with dilation values 2,3,4,5 to blocks; the number of filters in the convolutional frame layers C is set to 2048 equal to the number of filters in the bottleneck of the SE-Res2Net Block; ASP is used; embedding layer size is set to 512.

We use the Adaptive Curriculum Learning loss function with m and s being fixed to 0.35 and 32 respectively during the whole training.

Domain Generalized SE-ResNet34 The Domain agnostic extractor construction is inspired by the work [18] with some minor changes. First of all, we use the original language labels. The base architecture is modified to a commonly used ResNet-like SR system. As a frame-level network SE-ResNet34 with the 2 times reduced base width (32) is used. It is followed by ASP layer the output of which is passed to the two parallel Maxout-based [19] segment-level networks. The first stream responsible for the speaker discrimination is passed to the angle-linear head returning the vector of speaker probabilities. The second stream is doing the domain discrimination, it is comprised of 2-hidden dense layers and one classification layer, returning a domain probability. We use gradient reversal [20] to subtract the gradients of the domain discriminator from the embedding layer.

The Domain Generalized SE-ResNet34 (DG-SE-ResNet34) model is trained by the multi-task learning as in [18] and is forced to learn speaker embeddings such that they would contain less domain information. Multi-task learning is based on three tasks, the first is SR done by SC-AAM-Softmax (margin=0.2, scale=30, number of sub-classes=2), second is domain discrimination done by cross-entropy loss and the third is the entropy minimization [21, 22] of the SR predictions done by entropy loss [18], which forces the model to perform low-density separation between different speakers.

2.3. αQE for Embedding Aggregation

There are three main approaches to aggregate multiple enrollment utterances into one vector. The first is to simply do the score-level aggregation, by which we first compare each enrollment embedding with the test embedding independently, and then take the average of those scores with the unit weights. The second and third approaches are doing the embedding-level fusion. Embeddings can be aggregated simply by arithmetic mean, as well as by weighted mean. Weighting in our experiments is done by the αQE [23] inspired approach, which is strongly dependent on cosine distance to the test embedding. Our method αQE_p is different from the original αQE approach by the notion of normalized weights. Bellow, we demonstrate the pseudo-code for the αQE_p-based average enrollment vector computation.

Algorithm 1 αQE_p

Input: $(t, \{x_i, \dots, x_c\})$ - trial; t - test acoustic feature vector, x_i - enrollment acoustic feature vector; f - feature extractor (extracts speaker embedding);

1: Set optimal $\alpha \in [0; +\infty)$
2: Compute weights w_i for each pair (t, x_i):
3: **for** $x_i = x_1, x_2, \dots, x_c$ **do**
4: $w_i = (\frac{\mathbf{f(x_i)}}{\|\mathbf{f(x_i)}\|_\mathbf{2}^\mathbf{2}})^T (\frac{\mathbf{f(t)}}{\|\mathbf{f(t)}\|_\mathbf{2}^\mathbf{2}})$ ▷ cosine score
5: $w_i' = (\frac{(w_i + 1)}{2})^\alpha$ ▷ min-max-norm. & apply α
6: $\widetilde{w_i'} = \frac{w_i'}{\sum_{j=1}^{c} w_j'}$ ▷ normalize
7: Assert ($\sum_{i=1}^{c} \widetilde{w_i'} = 1$);
8: $v = \sum_{i=1}^{c} w_i' * \mathbf{f(x_i)}$ ▷ mean enrollment vector

From the above equations, it can be seen that when $\alpha = 0$ the v is turned to the average embedding, with the αQE turning to Average Query Expansion (AQE). We could also try to omit min-max normalization from the proposed αQE_p as in the original αQE scheme and denote this scheme as αQE_n. The experimental results are given in the latter section.

2.4. K-reciprocal-based Trail Classifier (KrTC)

To predict either the trial pairs contains imposters or targets we have derived the k-reciprocal set [24] for the test embedding taking the enrollment probe in this trial for the gallery to search reciprocals over. And used the size of this reciprocal set as a predictor of impostor/target pairs. Then, we used this prediction to sort the enrollment vectors in the order of relevance of enrollment vectors to the test, where the relevance for imposters is the small cosine score between test and an enrollment vector and inversely large cosine score for the targets. Bellow in Algorithm 2, there is a pseudo-code for the $KrTC$ algorithm to classify trials and apply relevance sorting to enrollment embeddings.

As an equation (1) in the Algorithm 2 states the enrollment embedding is named to be the reciprocal for the test embedding if the K-NN neighbourhood of the enrollment vector computed over the trial contains the test embedding, subject to the test embedding also containing this enrollment embedding in it's K-NN neighbourhood.

2.5. Loss Function

Curriculum Learning Loss For embedding extractors training the novel Adaptive Curriculum Learning Loss approach [25] was used. The main idea of this training strategy is to add to Additive Angular Margin Loss a learnable parameter t, which

Algorithm 2 $KrTC$

Input: $(t,\ X)$ - trial; t - test embedding, $X = \{x_i, \ldots, x_c\}$ and x_i - enrollment embedding;
Hyperparameters: $threshold$ - used to decide whether the trial is target or impostor, K_p - the fraction of enrollment set to use for K-NN computation.
Notation:$N(t, K)$-the K-NN for t over all embeddings in a trial

function KRTC(t, X, $threshold$, K_p):
    $c = len(X)$ ▷ enrollment set size
    $K = K_p * c$ ▷ number of nearest neighbours
    if $c > K + 1$ **then** ▷ sufficient for $t \notin N(x_j, 0.9c)$

$$R_t = \{x_j | (x_j \in N(t, K) \wedge t \in N(x_j, K))\} \quad (1)$$

        $p_t = \dfrac{len(R_t)}{c}$ ▷ fraction of k-reciprocals
        if $p_t < threshold$ **then** ▷ impostor
            X = SORT(X, t) ▷ smallest scores first
        else ▷ target
            X = SORT(X, t)[::-1] ▷ biggest scores first
    return X
function SORT(X, t): ▷ in the increasing order of scores
    scores = []
    for x_i in X **do**
        scores[i] = $cosine\_score(X, t)$
    $inds$ = $argsort$(scores)
    return $X[inds]$

assigns different importance to semi-hard and hard samples at different training stages. Main features of the loss function formulated in equation (2)

$$L = -log\frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1, j\neq y_i}^{n} e^{sN(t^{(k)}, \cos(\theta_j))}},$$
$$N(t, cos_{\theta_j}) = \begin{Bmatrix} cos\theta_j, T(cos\theta_{y_i}) - cos\theta_j \geqslant 0 \\ cos\theta_j(t + cos\theta_j), T(cos\theta_{y_i}) - cos\theta_j < 0 \end{Bmatrix}, \quad (2)$$

3. Datasets

Training dataset We use concatenated VoxCeleb1 and VoxCeleb2 (SLR47) [26] LibriSpeech [27], Mozilla Common Voice Farsi [28] and DeepMine [29, 30] (Task 2 Train Partition). For VoxCeleb1 and VoxCeleb2 datasets, for each video, we concatenated all it's files into one chunk. For LibriSpeech and DeepMine we concatenated segments randomly into 20s chunks and for Mozilla Common Voice Farsi into 60s chunks. The overall number of speakers in the resulting set is 11939. 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]. The resulting files were processed according to subsection 2.1.

Development and evaluation dataset The enrollment data in SdSVC text independent task [31] consists of one or several variable-length utterances for each speaker with speech duration for each file roughly 4 to 180 seconds. The duration of the test utterances varies between 1 to 8 seconds. The development set is a subset of main evaluation data.

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

4. Experiments

All models in this work were trained using the PyTorch framework.

4.1. ResNet

We use SGD optimizer with momentum 0.9 and weight decay 10^{-5} for the head and 10^{-6} for the rest of the layers. OneCycleLR scheduler with the maximum learning rate fixed to 2 is also used. AMP (Automatic Mixed Precision) with half precision was used for increasing batch size per GPU and training speed. All models have been trained for 15 epochs on randomly sampled 4-sec crops of each train dataset utterance. The results obtained by ResNet models can be viewed in Table 2.

4.2. ECAPA-TDNN

We use SGD optimizer with momentum 0.9 and weight decay 10^{-4} for the head and 10^{-5} for the rest of the layers. OneCycleLR policy with the maximum learning rate value 0.4 is used to scheduler learning rate during training. All models have been trained with AMP for 10 epochs on 2-sec random crops of each training dataset utterance. The results obtained by the ECAPA-TDNN model can be viewed in Table 2.

4.3. Domain Generalized SE-ResNet34 & Language Classifier

As for the domain labels we have used language labels depending on the dataset the utterance is drawn from: **English** for VoxCeleb1, VoxCeleb2, Deepmine English, portions of Librispeech; **Persian** for Mozila Farsi, Deepmine Farsi. We ignore the fact that the part of the VoxCeleb 2 contains a variety of languages and set the domain labels of all VoxCeleb to English. As an optimizer, we took Adam with a fixed learning rate schedule and learning rate being set to 0.001. We also used soft speaker labels [32] with the smoothing parameter set to 0.1, SpecAugment [33] frequency and time masking technique with the number of masks set to 1 and maximum frequency and time mask widths set to 10. As for the input size, we iteratively fine-tuned the network with random 2-6 second crops. Also, we have used inverse class frequency sampling to balance the counts of both language utterances.

4.4. Scoring

As both development and test protocols are comprised of multi-enroll trials and not all enrollment samples may be equally representative of the speaker uttering them we use embedding aggregation technique with $KrTC$ from subsections 2.3 and 2.4.

α **QE** We have experimented with not using the whole set of enrollment vectors in a trial, but rather take top n% of them that are close to the test. This way we were able to remove enrollment outliers out of the set. The results of grid-search on development set over different combinations of weighting variants of αQE and α values are given in Table 3.

KrTC As long as we have noticed the positive effect of using only the portion of enrollment vectors (top $n\%$) in a trial to get the average enrollment vector, we have assumed that it would be beneficial to do the selection of those top $n\%$ of enrolls taking into account whether the whole enrollment set is impostor or target to the given test. As a predictor, we have used $KrTC$ introduced in subsection 2.4 with $threshold = 0.2$ and $K_p = 0.9$ (i.e. 90%), which sorts the enrollment vectors by their scores with test based on the predicted type of the trial (targets - decreasing order, impostors - increasing order). To de-

Table 2: *The results of models on VoxCeleb1 and SdSVC datasets.*

Models	Params	Inference time (RT)	VoxCeleb1		SdSVC dev		SdSVC eval
			EER(%)	minDCF0.01	EER(%)	minDCF0.01	minDCF0.01
ResNetNAS	33.4M	2.80	1.03	0.083	2.11	0.127	0.078
ResNet101-like	33.4M	2.78	1.09	0.095	2.16	0.173	—
ECAPA-TDNN	77.3M	4.92	1.51	0.157	2.15	0.282	0.120
DG-SE-ResNet34	15.5M	12.58	1.27	0.181	3.737	0.229	—

Table 3: *Exploration of the best value of α on SDSVC 2021 dev for the αQE_p and αQE_n weighting scheme with top 90%.*

System	α	**EER** (αQE_p / αQE_n)	**minDCF 0.01** (αQE_p / αQE_n)
ResNet	0	2.188 / 2.188	0.131 / 0.131
	1	2.014 / 2.147	0.127 / **0.106**
	3	1.975 / 2.08	0.115 / 0.114
	4	**1.958** / 2.169	**0.110** / 0.120
ECAPA	0	2.160 / 2.160	0.281 / 0.281
	1	1.907 / 1.839	0.267 / 0.236
	5	1.839 / 1.920	0.230 / **0.198**
	6	**1.798** / 1.975	0.223 / 0.199

termine the optimal decision threshold and K_p values we used the SdSVC development set.

From lines 3 and 4 in $KrTC$ algorithm 4 it can be noticed that sorting by the k-reciprocal-based prediction can be done only under the condition $ifc > len(X) * 0.9 + 1$, which implies that the size of the enrollment set has to be greater than 10 (i.e $c > 10$), otherwise test embedding would be forced to be in the set of K-NN for all enrollment vectors (not enough vectors to compute KNN over). Table 4 demonstrates the results of searching for the optimal top n% value for the trials with less than 10 enrollment vectors, while the top $n\%$ for the $c > 10$ trials being fixed to 90%.

Table 4: *KrTC with αQE_p on SdSVC 2021 dev.*

System	α	top n%	**EER**	**minDCF 0.01**
ResNet	4	50	1.646	0.109
	6	50	1.654	0.108
	7	50	1.648	**0.107**
	8	50	**1.626**	**0.107**
ECAPA	4	70	**1.682**	0.204
	5	70	1.694	0.200
	6	70	1.771	0.195
	9	70	1.771	**0.191**

4.5. Fusion

To ensemble ResNet and ECAPA systems we use the following fusion technique:

- **Fusion 1**: ResNet+ECAPA. Concatenate ResNet embeddings with DGResNet embeddings and ECAPA embeddings with ResNet embeddings, perform αQE with $\alpha = 6$ for ResNet and $\alpha = 3$ for ECAPA-TDNN concatenated embeddings, then compute cosine scores for them independently. Average the scores with the weights 0.05 and 0.95 (smaller weight to the fusion with DGResNet as it is of the worst performance).
- **Fusion 2**: ResNet+ECAPA+k-reciprocal. Apply $KrTC$ to ResNet and ECAPA embeddings independently with the parameters from subsection 4.4. If $c < 10$ take top 50% for ResNet and top 70% for ECAPA, else take top 90%. Perform αQE with $\alpha = 8$ for ResNet and $\alpha = 9$ for ECAPA embeddings. We find these parameters optimal for a unique system on SdSVC development set in terms of minDCF. Compute cosine scores. Average the scores for the embeddings of both models with equal weights.

5. Results and discussion

The results of our systems on the SdSVC evaluation dataset are presented in Table 5. The ResNet-based model show better quality compared to the ECAPA-TDNN model. αQE_p helps to improve the verification quality of the models on an evaluation set. This effect is stronger for aggregation of speaker embeddings computed by ECAPA-TDNN model. Our experiments with ECAPA-TDNN and ResNet-like architectures show that they can be successfully fused due to the significant difference in architectures. K-reciprocal can help to improve results of both single and score-level fusion-based systems.

Table 5: *Systems on SdSVC 2021 test with αQE_p weighting scheme.*

System	α	**minDCF 0.01**
ECAPA	0	0.120
ECAPA	3	0.079
ECAPA + k-recip	9	0.077
ResNet	0	0.078
ResNet	8	0.067
ResNet+k-recip	8	0.069
Fusion 1	6 / 3	**0.058**
Fusion 2	8 / 9	0.063

6. Conclusions

Results of the competition confirm that deep ResNet and ECAPA-TDNN architectures are robust and allow to obtain a significantly better quality of speaker verification for short-duration utterances compared with the baseline TDNN method. Obtained results confirm that ResNet-NAS architectures allow improving the quality of speaker verification compared to the basic ResNet architecture with the same number of parameters. The proposed α QE multi-enroll aggregation technique can improve the quality of speaker recognition according to results on SdSVC development and evaluation datasets.

7. Acknowledgements

This research was financially supported by ITMO University.

8. References

[1] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "Voxsrc 2020: The second voxceleb speaker recognition challenge," 2020.

[2] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The interspeech 2020 far-field speaker verification challenge," 2020.

[3] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "SdSV Challenge 2020: Large-Scale Evaluation of Short-Duration Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 731–735. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1485

[4] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In Defence of Metric Learning for Speaker Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2977–2981. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1064

[5] X. Xiang, "The xx205 system for the voxceleb speaker recognition challenge 2020," 2020.

[6] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," 2017.

[7] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxceleb speaker recognition challenge 2020 system description," 2020.

[8] 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.

[9] 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.

[10] 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.

[11] A. Gusev, V. Volokhov, A. Vinogradova, T. Andzhukaev, A. Shulipa, S. Novoselov, T. Pekhovsky, and A. Kozlov, "Stc-innovation speaker recognition systems for far-field speaker verification challenge 2020," in *INTERSPEECH*, 2020.

[12] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," 2019.

[13] X. Xiang, "The xx205 system for the voxceleb speaker recognition challenge 2020," 2020.

[14] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 558–567.

[15] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," 2021.

[16] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2650

[17] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-993

[18] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," 2019.

[19] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," 2013.

[20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 2014.

[21] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5031–5040.

[22] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," 2016.

[23] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2019.

[24] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3652–3661.

[25] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. xin Li, J. Li, and F. Huang, "Curricularface: Adaptive curriculum learning loss for deep face recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5900–5909, 2020.

[26] 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.

[27] 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)*, 2015, pp. 5206–5210.

[28] 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," 2020.

[29] H. Zeinali, H. Sameti, and T. Stafylakis, "DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English." in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 386–392.

[30] H. Zeinali, L. Burget, and J. Cernocky, "A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: the DeepMine database," in *Proc. ASRU 2019 The 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.

[31] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "Short-duration speaker verification (sdsv) challenge 2021:the challenge evaluation plan," arXiv preprint arXiv:1912.06311, Tech. Rep., 2020.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[33] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680