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

Delving into VoxCeleb: environment invariant speaker recognition

Joon Son Chung[1,2]*, *Jaesung Huh*[2]*, *Seongkyu Mun*[2]

[1]Visual Geometry Group, University of Oxford, UK
[2]Naver Corporation, South Korea

joon@robots.ox.ac.uk

Abstract

Research in speaker recognition has recently seen significant progress due to the application of neural network models and the availability of new large-scale datasets. There has been a plethora of work in search for more powerful architectures or loss functions suitable for the task, but these works do not consider what information is learnt by the models, apart from being able to predict the given labels.

In this work, we introduce an environment adversarial training framework in which the network can effectively learn speaker-discriminative and environment-invariant embeddings without explicit domain shift during training. We achieve this by utilising the previously unused 'video' information in the VoxCeleb dataset. The environment adversarial training allows the network to generalise better to unseen conditions. The method is evaluated on both speaker identification and verification tasks using the VoxCeleb dataset, on which we demonstrate significant performance improvements over baselines.

1. Introduction

Deep learning has been pushing the state-of-the-art in many fields of research over the recent years. These architectures can simultaneously learn feature representation and decision framework from large labelled datasets, removing the need to handcraft features for any given problem. Such progress has been facilitated by the availability of large-scale datasets, such as ImageNet [1] for image classification, Labeled Faces in the Wild [2] for face recognition and VoxCeleb [3] for speaker recognition. However, the weakness of data-driven approaches is that it is not possible to define what information is learnt by the models during the training process – whether it is the useful information or undesirable biases that are present in the dataset.

In speaker recognition, the challenge comes down to the ability to separate the voice characteristics and the environments in which the person's voice is recorded. The VoxCeleb dataset contains recordings from diverse but finite environments for each speaker, making it possible for the model to overfit to the environment as well as the voice characteristics. In order to prevent this, we must look beyond classification accuracy as the only learning objective.

In this paper, we propose a new framework for learning effective speaker embeddings at the same time as removing undesirable sources of variation such as environment information. This work is inspired by domain adaptation approaches [4, 5, 6, 7] and an extension of this work to bias removal in models [8]. Also of relevance are recent works [9, 10, 11] that have used adversarial training in speaker recognition for adaptation between distinct domains. A more detailed overview of these work will be given in Section 1.1.

In contrast to the previous work on domain adaptation, our model is trained to be invariant to environments and recording conditions *without* explicit domain shift or the use of domain annotation during training. The model is trained on the VoxCeleb dataset alone and there is no supervisory requirement beyond what is provided in the dataset. The network trained using the proposed framework generalises better to both unseen samples of seen speakers for speaker identification and to unseen speakers for speaker verification, as well as to unseen environments.

The paper is organised as follows. In Section 2, we discuss the adversarial learning framework that allows speaker representations to be trained whilst removing environment information without explicit labels. Section 3 describes the trunk architectures for the network and the dataset used for training. In Section 4, we demonstrate that the speaker recognition networks trained using the proposed framework yields significant improvements over baselines. The experiment on 'replayed' VoxCeleb dataset show that the gains are more pronounced in environments not seen during training. We also probe the trained network to find that much of the environment information has indeed been removed from the embedding.

1.1. Related works

Although this paper focuses on adversarial training, there has been a long history of research on methods to train noise and environment robust speaker embeddings, rang-

* These authors contributed equally to this work.

10.21437/Odyssey.2020-49

ing from pre-processing to data augmentation. Each of these will be described in the following paragraphs.

Traditional methods. Traditional literature on speaker recognition [12] have used speaker-dependent GMM mean components as speaker representation features, which are obtained by adapting the UBM to the speaker's voice. However, the adaptation, usually maximum a posteriori (MAP), adapts not only to speaker-specific characters of speech, but also to channel and other nuisance factors. To address this issue, joint factor analysis [13] and i-vector based approach [14], which decomposes speaker and channel components using eigenvoice or total variability matrix, have been used. However, in environment robust research [15, 16], these techniques have not been commonly applied to deep learning based methods since they require direct update of model statistics such as mean and covariance of GMM.

As reported in [17, 18], the use of mid-level features of DNN has generally yielded higher performance compared to the i-vector framework, when large training database can be utilised.

Pre-processing. DNN-based speech enhancement has been a popular field of research in the recent years, using generative adversarial networks (GAN) [19] or Wave-U-Net [20]. A recent work has analysed the effects of speech enhancement to speaker recognition [21], and [22] has proposed joint training speech enhancement network together with speaker recognition system in order to improve its robustness. Whilst there is a difference between the speaker recognition performance and perceptual signal quality, this work shows the potential of the joint training to improve the overall performance.

Data Augmentation. Data augmentation has been used widely in machine learning to improve the performance of systems by artificially increasing the amount and diversity of training data. This technique was first popularised in computer vision [23, 24] by using label preserving transformations to augment image data. The method has been adopted for speaker recognition [18] by adding noise [25] and reverberations [26] to the speech signal. This recipe has been used by many of the recent work in the field [27, 28, 29].

Adversarial training. Deep learning approaches have made breakthroughs in performance for many applications of machine learning, but the trained models often generalise poorly to unseen data. Recent works [4, 6] on domain adaptation have proposed to address this issue by introducing *domain confusion loss* in addition to the standard classification loss. The domain confusion loss tries to match the distribution between source and target domains in order to confuse the classification layers, so that the samples from both domains are indistinguishable for the classifier.

Similar approaches have been adopted in speaker recognition to adapt speaker embeddings to a new domain. [9, 10, 11] have used domain adaptation training similar to [4, 6] between languages [9, 10, 11] and between datasets [10, 30].

Of closest relevance to our work is [8] that extends the work on domain adversarial training to *bias removal* for face recognition task by introducing auxiliary classifiers and confusion losses for undesirable sources of variations (e.g. age, nationality) that the primary classifier (identity) should be invariant to. [31] have replicated similar strategy for speaker recognition task, where the auxiliary classes are the types of noise that have been added to the speech signal.

The commonality amongst the previous work, on speaker recognition or otherwise, is that they require explicit domain or class labels in the source of variation that the trained embedding should be invariant to, and this must be a well-defined category. In contrast, our method is capable of learning environment-robust embeddings without explicit domain shift or auxiliary class labels.

2. Learning framework

In this section, we propose a training framework that aims to learn a feature representation that encapsulates useful speaker information, whilst being uninformative for undesirable sources of variations such as environment or recording conditions.

An overview of our framework is given in Figure 2. The speaker network has a single classification loss, which classifies the audio segment into one of 1,211 speakers. The environment network is first trained to determine whether or not the two audio segments are from the same environment, and also used to remove this information from the speaker embedding.

2.1. Batch formation

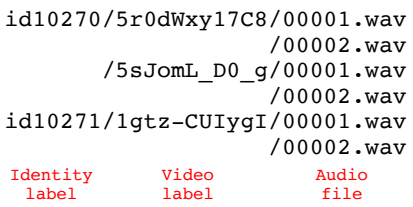

Figure 1: Data labels in the VoxCeleb dataset.

Each minibatch consists of three 2-second audio segments from N different speakers. Two of the three audio segments from each speaker are from the same video, and the other is from a different video. The two segments can be from different parts of the same audio clip if video labels are not available. The video reference can be inferred from the file path in the VoxCeleb dataset, as shown in

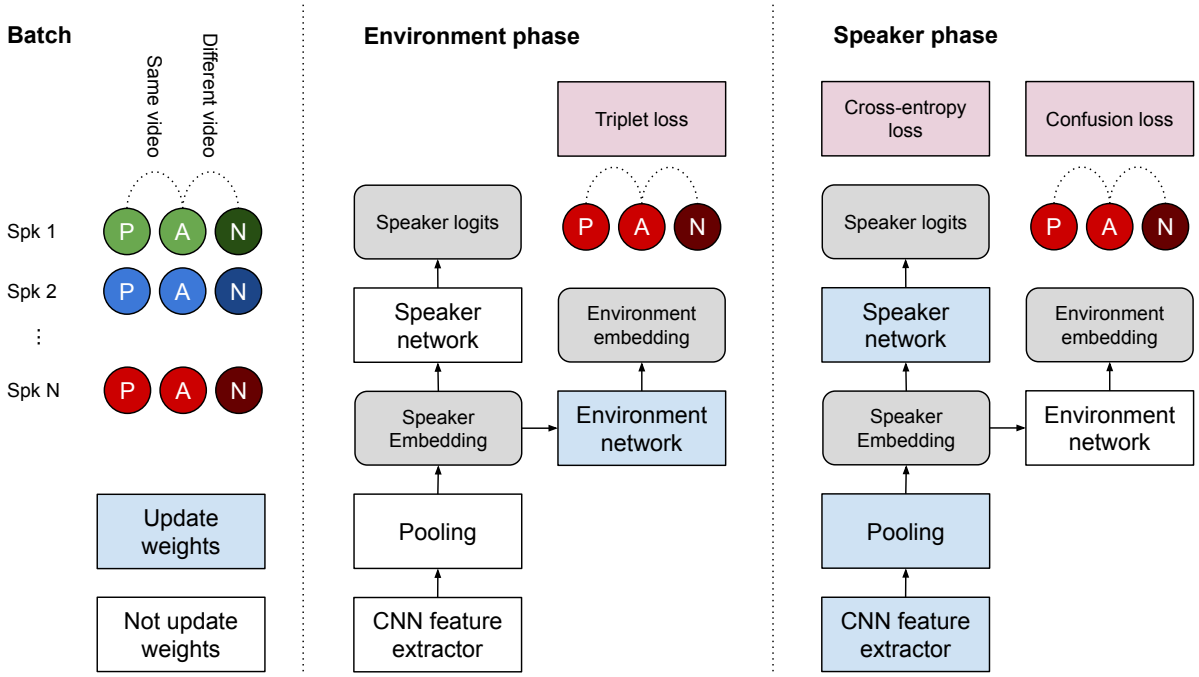

Figure 2: Overview of the training strategy. 'Confusion loss' minimises the KL divergence between the softmax of the triplet distances and a uniform distribution. **P** (positive) and **A** (anchor) are from the same video, **N** (negative) is from a different video from the anchor.

Figure 1. Here, the assumption is that the pair of clips from the same video would have been recorded in similar environments, whereas the clips from different videos would have more different channel characteristics. An example of such input batch is depicted in left hand column of Figure 2.

2.2. Algorithm

The algorithm is described by the pseudocode in Listing 1 and described below.

The speaker embedding is first extracted by the CNN feature extractor, and pooled over time using one of the two pooling strategies described in Section 3.2.

Environment phase. The environment network is trained to predict whether or not the input audio segments come from the same environment (same video) with a triplet loss, the anchor and a segment from the same video as the anchor forming the positive pair, and the anchor and a segment from a different video forming a negative pair. Suppose x_a, x_p, x_n are anchor, positive and negative speaker input representations and e_a, e_p, e_n are the corresponding outputs of the environment network. Then, environment loss L_e is

$$L_e = max(0, ||e_a - e_p||_2^2 - ||e_a - e_n||_2^2 + m) \quad (1)$$

where m is a margin of triplet loss. The gradient is back-propagated only to the environment network, so the CNN feature extractor is not optimized during this phase.

Speaker phase. The CNN feature extractor and the speaker recognition network are trained simultaneously using the standard cross-entropy loss. In addition, the confusion loss penalises the network's ability to discriminate between the clips that originate from the same environment (same video) and those from different environments – this is done by minimising the entropy between the softmax of the triplet distances and a uniform distribution. The environment or channel information can be seen as undesirable sources of variations that should be absent from an ideal speaker embedding.

The loss function in this phase is computed as follows. Let $\mathbf{s} = \{s_a, s_p, s_n\}$ be the corresponding outputs of the speaker network and y be the speaker label of $\mathbf{s}$. The loss function L_s is

$$\begin{aligned} p_{dist} &= softmax(||e_a - e_p||_2^2, ||e_a - e_n||_2^2) \\ L_s &= CE(\mathbf{s}, y) + \alpha * KL(p_{dist}||p_{unif}) \end{aligned} \quad (2)$$

where p_{unif} is an uniform distribution. Here, $CE(\mathbf{s}, y)$ is the cross entropy loss between speaker logits $\mathbf{s}$ and label y . $KL(p_a||p_b)$ is the KL divergence between distribution p_a and p_b. The extent to which the confusion loss contributes to the overall loss function is controlled by the variable α (e.g. 0, 1, 10, 30).

```
## Let xa, xp, xn, y be anchor, positive, negative speaker inputs and class labels.
## netcnn is the CNN feature extractor, netspk and netenv are fully connected networks.

optimizer       = optim.SGD([netcnn.parameters(), netspk.parameters()]);
disc_optimizer  = optim.SGD(netenv.parameters());

for xa, xp, xn, y in loader:
    ca, cp, cn      = netcnn.forward(xa), netcnn.forward(xp), netcnn.forward(xn)

    # Environment phase
    ea_, ep_, en_   = netenv.forward(ca.detach()), netenv.forward(cp.detach()), netenv.forward(cn.detach())
    disc_loss       = torch.mean(F.relu(torch.pow(l2_dist(ea_, ep_)) - torch.pow(l2_dist(ea_, en_)) + margin))
    disc_loss.backward();
    disc_optimizer.step();

    # Speaker phase
    sa, sp, sn      = netspk.forward(ca), netspk.forward(cp), netspk.forward(cn)
    ea, ep, en      = netenv.forward(ca), netenv.forward(cp), netenv.forward(cn)
    triplet_logits  = F.log_softmax(torch.stack((l2_dist(ea, ep),l2_dist(ea, en)),dim=1))
    x               = torch.cat((sa,sp,sn),dim=0)
    loss            = nn.CrossEntropyLoss(x, y.repeat(3)) + alpha * nn.KLDivLoss(triplet_logits, target=uniform)
    loss.backward();
    optimizer.step();
```

Listing 1: PyTorch-style pseudocode for the training scheme

3. Models and Experiments

This section describes the network architecture, the dataset and details of the experiments.

3.1. CNN feature extractor

Experiments are performed on two different architectures and in Table 1.

VGG-M-40. The original VGG-M model has been proposed for image classification [32] and adapted for speaker recognition by [3]. Whilst it is not a state-of-the-art network, the network is known for high efficiency and good classification performance. VGG-M-40 is a further modification of the network used by [3] to take 40-dimensional filterbanks as inputs instead of the 513-dimensional spectrogram, significantly reducing the number of computations.

Thin ResNet-34. Residual networks [33] are used widely in image recognition and has recently been applied to speaker recognition [34, 35, 36]. Thin ResNet-34 is the same as the original ResNet with 34 layers, except with only one-quarter of the channels in each residual block in order to reduce computational cost.

3.2. Temporal aggregation

Since we want the network to be invariant to temporal position but *not* frequency, [3] has proposed aggregation layers that are fully connected only along the frequency axis. This produces a $1 \times T$ feature map before the pooling layers, described in the following sections.

Temporal average pooling (TAP). The TAP layer simply takes the mean of the features along the time domain.

Self-attentive pooling (SAP). Unlike the TAP layer that equally pools the features over time, [34] introduces a self-attentive pooling layer to pay attention to the frames that are more informative for utterance-level speaker recognition. This is effectively a weighted mean of the features (Equation 3), where the weights w_t are given by Equation 4 and 5 where W, b and μ are learnable matrices or vectors. x_t are utterance-level feature maps along time domain and e is the final speaker embedding.

$$e = \sum_{t=1}^{T} w_t x_t \tag{3}$$

$$h_t = tanh(W x_t + b) \tag{4}$$

$$w_t = \frac{exp(h_t^T \mu)}{\sum_{t=1}^{T} exp(h_t^T \mu)} \tag{5}$$

3.3. Speaker and environment networks

The speaker network is a linear classifier that consists of a single fully connected layer with the output size equal to the number of speakers (1,211).

The environment network has two fully connected layers of size 512, each preceded by ReLU activation and batch normalisation.

3.4. Dataset

We train our models end-to-end on the VoxCeleb1 dataset. It contains more than 150,000 utterances from 1,251 speakers.

For **identification**, we only train on the *overlapping part* of the development sets for identification and verification, so that the models trained for identification can be used to evaluate verification. This makes speaker

layer name	VGG-M-40	Thin ResNet-34
conv1	$5 \times 7, 96$, stride 2 3×3, max pool, stride 1×2	$7 \times 7, 16$, stride 2 3×3, max pool, stride 2
conv2	$5 \times 5, 96$, stride 2 3×3, max pool, stride 2	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 3$, stride 1
conv3	$3 \times 3, 256$, stride 1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 4$, stride 2
conv4	$3 \times 3, 256$, stride 1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 6$, stride 2
conv5	$3 \times 3, 256$, stride 1 3×3, max pool, stride 2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$, stride 2
fc	$4 \times 1, 512$, stride 1	$9 \times 1, 512$, stride 1

Table 1: Modified VGG-M and ResNet architectures. ReLU and batchnorm layers are not shown. Each row specifies the number of convolutional filters, their sizes and strides as **size $\times$ size, # filters, stride**. The output from the fully connected layer is ingested by the pooling layers.

identification a 1,211-way classification task, and the test set consists of unseen utterances of seen speakers during training.

For **verification**, all speech segments from the 1,211 development set speakers are used for training, and the trained model is then evaluated on the 40 unseen test set speakers. The statistics are summarised in Table 2.

Split	# Speakers	# Utterances
Dev (Iden.)	1,211	140,638
Test (Iden.)	1,211	7,972
Dev (Ver.)	1,211	148,610
Test (Ver.)	40	4,874

Table 2: Development and test splits for speaker identification and verification. Note that the identification split is different from that in the original VoxCeleb paper [3].

3.5. Training details

Input representations. During training, we use a fixed length 2 second temporal segment, extracted randomly from each utterance. Spectrograms are extracted with a hamming window of width 25ms and step 10ms. For ResNet, the 257-dimensional raw spectrograms are used as the input to the network. For VGG networks, 40-dimensional Mel filterbanks are used as the input. Mean and variance normalisation (MVN) is performed on every frequency bin of the spectrogram and filterbank at utterance-level. No voice activity detection (VAD) or data augmentation is used in training.

Speaker verification. The network has been trained for a n-way classification task, but the verification task requires a measure of similarity. The final layer in the classification network is replaced with one of output dimension 512, and this layer is re-trained with contrastive loss and hard negative mining. The preceding layers (i.e. the CNN feature extractor) is *not* finetuned with the contrastive loss, which is in line with the training procedure of [3, 35].

Implementation details. Our implementation is based on the PyTorch framework [37] and trained on NVIDIA Tesla P40 accelerators. The network is trained using Stocastic Gradient Descent (SGD) with an initial learning rate of 10^{-3}, decreasing by a factor of 0.95 every epoch. Batch normalisation [38] is used during training. The training is stopped after 100 epochs or whenever the validation error did not improve for 10 epochs, whichever is sooner.

4. Results

In this section, we first compare the performance of our method to baselines, and also probe the network to see if the environment information has been removed from the embedding.

4.1. Speaker recognition

The trained network is evaluated on the VoxCeleb1 test set. We sample ten 2-second temporal crops from each test segment, and compute the distances between all possible combinations ($10 \times 10 = 100$) from every pair of segments. The mean of the 100 distances is used as the score. This protocol is in line with that used by [35].

Table 3 reports results for multiple models used for evaluation. Across both speaker identification and verification tasks, the models trained with the proposed adversarial strategy ($\alpha > 0$) consistently outperform those trained without ($\alpha = 0$).

Model	Pooling	α	Iden. T1	Iden. T5	Ver. EER	Replay EER	Env. EER
VGG-M [3]	TAP	0	-	-	7.82%	-	-
VGG-M-40	TAP	0	67.62%	82.90%	8.44%	15.16%	**18.72%**
		1	72.74%	87.15%	8.15%	14.60%	20.01%
		10	75.96%	89.48%	7.79%	14.23%	21.93%
		30	**77.70%**	**90.29%**	**7.61%**	**13.21%**	23.87%
VGG-M-40	SAP	0	68.13%	83.67%	8.02%	15.74%	**18.45%**
		1	70.99%	85.64%	8.31%	14.68%	18.98%
		10	76.01%	89.53%	7.93%	14.49%	21.63%
		30	**77.31%**	**90.48%**	**7.82%**	**13.88%**	23.55%
Thin ResNet-34	SAP	0	83.68%	92.90%	5.71%	13.14%	**20.43%**
		1	88.34%	95.48%	5.38%	12.41%	23.04%
		10	**89.00%**	95.94%	**5.26%**	**10.56%**	25.74%
		30	**89.00%**	**96.15%**	5.37%	10.58%	26.38%

Table 3: Results on speaker identification and verification tasks. Models for both tasks have been trained *only* on VoxCeleb1. Note that the test set split for the identification task is different from [3]. **T1, T5:** top-1 and top-5 accuracies; **EER:** Equal Error Rate.

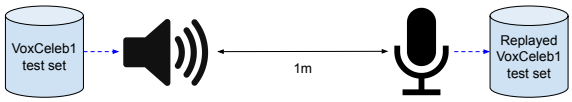

Figure 3: Setup for the replay experiment.

4.2. Replay experiment

The 'replay' experiment measures the performance on the same VoxCeleb test set used in the speaker verification task, but played through a reference speaker and re-recorded using a Jabra SPEAK 510 microphone. This results in a significant change in channel characteristics and deterioration of sound quality. The models are identical to those used in previous experiments, and not fine-tuned on the replayed segments.

The results are shown in the column of Table 3 named **Replay EER**. The improvement in performance as α increases is more pronounced in this setting (24% relative improvement in EER for the ResNet model compared to 9% improvement in the original VoxCeleb dataset), which suggests that the model trained with the proposed adversarial training generalises much better to unseen environments or channels.

4.3. Analysis on the removal of environment information

We perform experiments to verify that the adversarial training helps to remove environment information from the embedding. The test list for evaluating the environment recognition consists of 9,486 same speaker pairs, half of which come from the same video and the other half from different videos. We report the results in the right-most column of Table 3. A lower EER indicates that the network is better at predicting whether or not a pair of audio segments come from the same video. The results demonstrate that environment recognition performance decreases with the increase of α, which shows that the unwanted environment information has indeed been removed from the speaker embedding to an extent.

5. Conclusion

In this paper, we have proposed an environment adversarial training framework in which the network can learn speaker-discriminative and environment-invariant embeddings. We have evaluated the proposed method on both speaker identification and verification tasks using the VoxCeleb dataset, on which the performance of the method exceeds that of baselines by a significant margin on an unseen test set. We also probe the trained network to verify that much of the environment information has indeed been removed from the embedding.

Acknowledgements. We would like to thank Triantafyllos Afouras, Jisu Choi, Sanghyuk Chun, Hee Soo Heo and Bong-Jin Lee for helpful comments.

6. References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, 2007.

[3] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[5] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 469–477.

[6] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 4068–4076.

[7] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[8] M. Alvi, A. Zisserman, and C. Nellaker, "Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings," in *Workshop on Bias Estimation in Face Analytics, ECCV*, 2018.

[9] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6006–6010.

[10] G. Bhattacharya, J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6041–6045.

[11] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6226–6230.

[12] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[13] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[14] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[15] T. Hasan and J. H. Hansen, "Maximum likelihood acoustic factor analysis models for robust speaker verification in noise," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 2, pp. 381–391, 2013.

[16] Y. Lei, L. Burget, and N. Scheffer, "A noise robust i-vector extractor using vector taylor series for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6788–6791.

[17] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 4814–4818.

[18] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 5329–5333.

[19] M. H. Soni, N. Shah, and H. A. Patil, "Time-frequency masking-based speech enhancement using generative adversarial network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 5039–5043.

[20] R. Giri, U. Isik, and A. Krishnaswamy, "Attention wave-u-net for speech enhancement," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2019, pp. 249–253.

[21] O. Novotnỳ, O. Plchot, O. Glembek, L. Burget, et al., "Analysis of dnn speech signal enhancement for robust speaker recognition," *Computer Speech & Language*, vol. 58, pp. 403–421, 2019.

[22] S. Shon, H. Tang, and J. Glass, "Voiceid loss: Speech enhancement for speaker verification," in *Interspeech*, 2019.

[23] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1106–1114.

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

[26] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 5220–5224.

[27] H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, "Speaker augmentation and bandwidth extension for deep speaker embedding," in *Interspeech*, 2019, pp. 406–410.

[28] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," in *Interspeech*, 2019, pp. 4045–4049.

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

[30] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 4889–4893.

[31] J. Zhou, T. Jiang, L. Li, Q. Hong, Z. Wang, and B. Xia, "Training multi-task adversarial network for extracting noise-robust speaker embedding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6196–6200.

[32] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proceedings of the British Machine Vision Conference*, 2014.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[34] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Speaker Odyssey*, 2018.

[35] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[36] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, 2015.