VoxCeleb2: Deep Speaker Recognition

Joon Son Chung[†], *Arsha Nagrani*[†], *Andrew Zisserman*

Visual Geometry Group, Department of Engineering Science,
University of Oxford, UK

{joon,arsha,az}@robots.ox.ac.uk

Abstract

The objective of this paper is speaker recognition under noisy and unconstrained conditions.

We make two key contributions. First, we introduce a very large-scale *audio-visual* speaker recognition dataset collected from open-source media. Using a fully automated pipeline, we curate VoxCeleb2 which contains over a million utterances from over 6,000 speakers. This is several times larger than any publicly available speaker recognition dataset.

Second, we develop and compare Convolutional Neural Network (CNN) models and training strategies that can effectively recognise identities from voice under various conditions. The models trained on the VoxCeleb2 dataset surpass the performance of previous works on a benchmark dataset by a significant margin.

Index Terms: speaker identification, speaker verification, large-scale, dataset, convolutional neural network

1. Introduction

Despite recent advances in the field of speaker recognition, producing single, compact representations for speaker segments that can be used efficiently under noisy and unconstrained conditions is still a significant challenge. In this paper, we present a deep CNN based neural speaker embedding system, named VGGVox, trained to map voice spectrograms to a compact Euclidean space where distances directly correspond to a measure of speaker similarity. Once such a space has been produced, other tasks such as speaker verification, clustering and diarisation can be straightforwardly implemented using standard techniques, with our embeddings as features.

Such a mapping has been learnt effectively for face images, through the use of deep CNN architectures [1, 2, 3] trained on large-scale face datasets [4, 5, 6]. Unfortunately, speaker recognition still faces a dearth of large-scale freely available datasets in the wild. VoxCeleb1 [7] and SITW [8] are valuable contributions, however they are still an order of magnitude smaller than popular face datasets, which contain millions of images. To address this issue we curate VoxCeleb2, a large scale speaker recognition dataset obtained automatically from open-source media. VoxCeleb2 consists of over a million utterances from over 6k speakers. Since the dataset is collected 'in the wild', the speech segments are corrupted with real world noise including laughter, cross-talk, channel effects, music and other sounds. The dataset is also multilingual, with speech from speakers of 145 different nationalities, covering a wide range of accents, ages, ethnicities and languages.

We train VGGVox on this dataset in order to learn speaker discriminative embeddings. Our system consists of three main variable parts: an underlying deep CNN *trunk* architecture, which is used to extract the features, a pooling method which is used to aggregate features to provide a single embedding for a given utterance, and a pairwise loss trained on the features to directly optimise the mapping itself. We experiment with both VGG-M [9] and ResNet [10] based trunk CNN architectures.

We make the following four contributions: (i) we curate and release a large-scale dataset which is significantly larger than any other speaker verification dataset. It also addresses a lack of ethnic diversity in the VoxCeleb1 dataset (section 3); (ii) we propose deep ResNet-based architectures for speaker embedding suitable for spectrogram inputs (section 4); (iii) we beat the current state of the art for speaker verification on the VoxCeleb1 test set using our embeddings (section 5); and (iv) we propose and evaluate on a new verification benchmark test set which involves the entire VoxCeleb1 dataset.

The VoxCeleb2 dataset can be downloaded from http://www.robots.ox.ac.uk/~vgg/data/voxceleb2.

2. Related works

Traditional methods. Traditionally, the field of speaker recognition has been dominated by i-vectors [11], classified using techniques such as heavy-tailed PLDA [12] and Gauss-PLDA [13]. While defining the state-of-the-art for a long time, such methods are disadvantaged by their reliance on handcrafted feature engineering. An in-depth review of these traditional methods is given in [14].

Deep learning methods. The success of deep learning in computer vision and speech recognition has motivated the use of deep neural networks (DNN) as feature extractors combined with classifiers, though not trained end-to-end [15, 16, 17, 18, 19]. While such fusion methods are highly effective, they still require hand-crafted engineering. In contrast, CNN architectures can be applied directly to raw spectrograms and trained in an end-to-end manner. For example, [20] uses a Siamese feed-forward DNN to discriminatively compare two voices, however this relies on pre-computed MFCC features, whilst [21] also learns the features instead of using MFCCs. The most relevant to our work is [22], who train a neural embedding system using the triplet loss. However, they use private internal datasets for both training and evaluation, and hence a direct comparison with their work is not possible.

Datasets. Existing speaker recognition datasets usually suffer from one or more of the following limitations: (i) they are either obtained under controlled conditions (e.g., from telephone calls [23, 24] or acoustic laboratories [25, 26, 27]), (ii) they are manually annotated and hence limited in size [8], or (iii) not freely available to the speaker community [28, 27] (see [7] for a full review of existing datasets). In contrast, the VoxCeleb2 dataset does not suffer from any of these limitations.

[†] These authors contributed equally to this work.

Dataset	VoxCeleb1	VoxCeleb2
# of POIs	1,251	6,112
# of male POIs	690	3,761
# of videos	22,496	150,480
# of hours	352	2,442
# of utterances	153,516	1,128,246
Avg # of videos per POI	18	25
Avg # of utterances per POI	116	185
Avg length of utterances (s)	8.2	7.8

Table 1: *Dataset statistics for both* VoxCeleb1 *and* VoxCeleb2. *Note* VoxCeleb2 *is more than 5 times larger than* VoxCeleb1. *POI: Person of Interest.*

Dataset	Dev	Test	Total
# of POIs	5,994	118	6,112
# of videos	145,569	4,911	150,480
# of utterances	1,092,009	36,237	1,128,246

Table 2: *Development and test set split.*

3. The VoxCeleb2 Dataset

3.1. Description

VoxCeleb2 contains over 1 million utterances for over 6,000 celebrities, extracted from videos uploaded to YouTube. The dataset is fairly gender balanced, with 61% of the speakers male. The speakers span a wide range of different ethnicities, accents, professions and ages. Videos included in the dataset are shot in a large number of challenging visual and auditory environments. These include interviews from red carpets, outdoor stadiums and quiet indoor studios, speeches given to large audiences, excerpts from professionally shot multimedia, and even crude videos shot on hand-held devices. Audio segments present in the dataset are degraded with background chatter, laughter, overlapping speech and varying room acoustics. We also provide face detections and face-tracks for the speakers in the dataset, and the face images are similarly 'in the wild', with variations in pose (including profiles), lighting, image quality and motion blur. Table 1 gives the general statistics, and Figure 1 shows examples of cropped faces as well as utterance length, gender and nationality distributions.

The dataset contains both development (train/val) and test sets. However, since we use the VoxCeleb1 dataset for testing, only the development set will be used for the speaker recognition task (Sections 4 and 5). The VoxCeleb2 test set should prove useful for other applications of audio-visual learning for which the dataset might be used. The split is given in Table 2. The development set of VoxCeleb2 has no overlap with the identities in the VoxCeleb1 or SITW datasets.

3.2. Collection Pipeline

We use an automatic computer vision pipeline to curate VoxCeleb2. While the pipeline is similar to that used to compile VoxCeleb1 [7], the details have been modified to increase efficiency and allow talking faces to be recognised from multiple poses, not only near-frontal. In fact, we change the implementation of every key component of the pipeline: the face detector, the face tracker, the SyncNet model used to perform active speaker verification, and the final face recognition model at the end. We also add an additional step for automatic duplicate removal. This pipeline allows us to obtain a dataset that is five times the size of [7]. We also note that the list of celebrity names spans a wider range of nationalities, and hence unlike [7], the dataset obtained is multi-lingual. For the sake of clarity, the key stages are discussed in the following paragraphs:

Stage 1. Candidate list of Persons of Interest (POIs). The first stage is to obtain a list of POIs. We start from the list of people that appear in the VGGFace2 dataset [4], which has considerable ethnic diversity and diversity in profession. This list contains over 9,000 identities, ranging from actors and sportspeople to politicians. Identities that overlap with those of VoxCeleb1 and SITW are removed from the development set.

Stage 2. Downloading videos. The top 100 videos for each of the POIs are automatically downloaded using YouTube search. The word 'interview' is appended to the name of the POI in search queries to increase the likelihood that the videos contain an instance of the POI speaking, as opposed to sports or music videos.

Stage 3. Face tracking. The CNN face detector based on the Single Shot MultiBox Detector (SSD) [29] is used to detect face appearances on every frame of the video. This detector is a distinct improvement from that used in [7], allowing the detection of faces in profile and extreme poses. We used the same tracker as [7] based on ROI overlap.

Stage 4. Face verification. A face recognition CNN is used to classify the face tracks into whether they are of the POI or not. The classification network used here is based on the ResNet-50 [10] trained on the VGGFace2 dataset. Verification is done by directly using this classification score.

Stage 5. Active speaker verification. The goal of this stage is to determine if the visible face is the speaker. This is done by using a multi-view adaptation [30] of 'SyncNet' [31, 32], a two-stream CNN which determines the active speaker by estimating the correlation between the audio track and the mouth motion of the video. The method can reject clips that contain dubbing or voice-over.

Stage 6. Duplicate removal. A caveat of using YouTube as a source for videos is that often the same video (or a section of a video) can be uploaded twice, albeit with different URLs. Duplicates are identified and removed as follows: each speech segment is represented by a 1024D vector using the model in [7] as a feature extractor. The Euclidean distance is computed between all pairs of features from the same speaker. If any two speech segments have a distance smaller than a very conservative threshold (of 0.1), then the the speech segments are deemed to be identical, and one is removed. This method will certainly identify all exact duplicates, and in practice we find that it also succeeds in identifying near-duplicates, e.g. speech segments of the same source that are differently trimmed.

Stage 7. Obtaining nationality labels. Nationality labels are crawled from Wikipedia for all the celebrities in the dataset. We crawl for country of *citizenship*, and not *ethnicity*, as this is often more indicative of accent. In total, nationality labels are obtained for all but 428 speakers, who were labelled as unknown. Speakers in the dataset were found to hail from 145 nationalities (compared to 36 for VoxCeleb1), yielding a far more ethnically diverse dataset (See Figure 1 (bottom, right) for the distribution of nationalities). Note also the percentage of U.S. speakers is smaller in VoxCeleb2 (29%) compared to VoxCeleb1 (64%) where it dominates.

Discussion. In order to ensure that our system is extremely confident that a person has been correctly identified (Stage 4), and that are speaking (Stage 5) without any manual interference, we set conservative thresholds in order to minimise the number of false positives. Since VoxCeleb2 is designed primarily as a training-only dataset, the thresholds are less strict compared to those used to compile VoxCeleb1, so that fewer

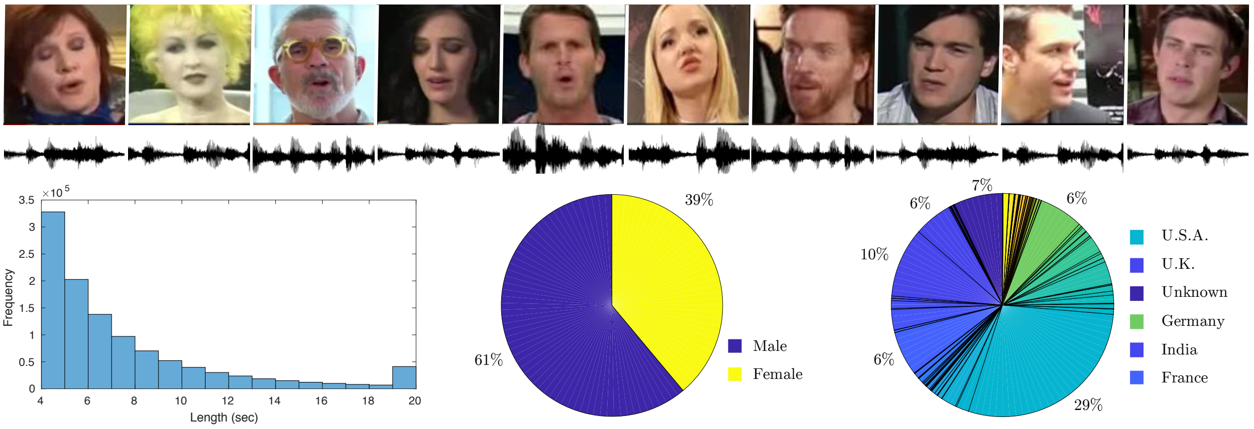

Figure 1: Top row*: Examples from the VoxCeleb2 dataset. We show cropped faces of some of the speakers in the dataset. Both audio and face detections are provided.* **Bottom row***: (left) distribution of utterance lengths in the dataset – lengths shorter than $20s$ are binned in $1s$ intervals and all utterances of $20s+$ are binned together; (middle) gender distribution and (right) nationality distribution of speakers. For readability, the percentage frequencies of only the top-5 nationalities are shown. Best viewed zoomed in and in colour.*

videos are discarded. Despite this, we have only found very few label errors after manual inspection of a significant subset of the dataset.

4. VGGVox

In this section we describe our neural embedding system, called VGGVox. The system is trained on short-term magnitude spectrograms extracted directly from raw audio segments, with no other pre-processing. A deep neural network trunk architecture is used to extract frame level features, which are pooled to obtain utterance-level speaker embeddings. The entire model is then trained using contrastive loss. Pre-training using a softmax layer and cross-entropy over a fixed list of speakers improves model performance; hence we pre-train the trunk architecture model for the task of identification first.

4.1. Evaluation

The model is trained on the `VoxCeleb2` dataset. At train time, pairs are sampled on-line using the method described in Section 4.3. The testing is done on the `VoxCeleb1` dataset, with the test pairs provided in that dataset.

We report two performance metrics: (i) the Equal Error Rate (EER) which is the rate at which both acceptance and rejection errors are equal; and (ii) the cost function

$$C_{det} = C_{miss} \times P_{miss} \times P_{tar} + C_{fa} \times P_{fa} \times (1 - P_{tar}) \quad (1)$$

where we assume a prior target probability P_{tar} of 0.01 and equal weights of 1.0 between misses C_{miss} and false alarms C_{fa}. Both metrics are commonly used for evaluating identity verification systems.

4.2. Trunk architectures

VGG-M: The baseline trunk architecture is the CNN introduced in [7]. This architecture is a modification of the VGG-M [9] CNN, known for high efficiency and good classification performance on image data. In particular, the fully connected *fc6* layer from the original VGG-M is replaced by two layers – a fully connected layer of 9×1 (support in the frequency domain), and an average pool layer with support $1 \times n$, where n depends on the length of the input speech segment (for example for a 3 second segment, $n = 8$). The benefit of this modification is that the network becomes invariant to temporal position but *not* frequency, which is desirable for speech, but not for images. It also helps to keep the output dimensions the same as those of the original fully connected layer, and reduces the number of network parameters by fivefold.

ResNets: The residual-network (ResNet) architecture [10] is similar to a standard multi-layer CNN, but with added skip connections such that the layers add residuals to an identity mapping on the channel outputs. We experiment with both ResNet-34 and ResNet-50 architectures, and modify the layers to adapt to the spectrogram input. We apply batch normalisation before computing rectified linear unit (ReLU) activations. The architectures are specified in Table 3.

layer name	res-34	res-50
conv1	$7 \times 7, 64$, stride 2	$7 \times 7, 64$, stride 2
pool1	3×3, max pool, stride 2	3×3, max pool, stride 2
conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
fc1	$9 \times 1, 512$, stride 1	$9 \times 1, 2048$, stride 1
pool_time	$1 \times N$, avg pool, stride 1	$1 \times N$, avg pool, stride 1
fc2	$1 \times 1, 5994$	$1 \times 1, 5994$

Table 3: *Modified Res-34 and Res-50 architectures with average pool layer at the end. ReLU and batchnorm layers are not shown. Each row specifies the number of convolutional filters and their sizes as* **size** $\times$ **size, # filters***.*

4.3. Training Loss strategies

We employ a contrastive loss [33, 34] on paired embeddings, which seeks to minimise the distance between the embeddings of positive pairs and penalises the negative pair distances for being smaller than a margin parameter α. Pair-wise losses such as the contrastive loss are notoriously difficult to train [35], and hence to avoid suboptimal local minima early on in training, we proceed in two stages: first, pre-training for identifcation using a softmax loss, then, second, fine-tuning with the contrastive loss.

Pre-training for identification: Our first strategy is to use softmax pre-training to initialise the weights of the network. The cross entropy loss produces more stable convergence than the contrastive loss, possibly because softmax training is not impacted by the difficulty of pairs when using the contrastive loss. To evaluate the identification performance, we create a held-out validation test which consists of all the speech segments from a single video for each identity.

Learning an embedding with contrastive loss – hard negative mining: We take the model pre-trained on the identification task, and replace the 5994-way classification layer with a fully connected layer of output dimension 512. This network is trained with contrastive loss.

A key challenge associated with learning embeddings via the contrastive loss is that as the dataset gets larger, the number of possible pairs grows quadratically. In such a scenario, the network rapidly learns to correctly map the easy examples, and hard negative mining is often required to improve performance to provide the network with a more useful learning signal. We use an offline hard negative mining strategy, which allows us to select harder negatives (*e.g.* top 1-percent of randomly generated pairs) than is possible with online (in-batch) hard negative mining methods [36, 35, 37] limited by the batch size. We do not mine hard positives, since false positive pairs are much more likely to occur than false negative pairs in a random sample (due to possible label noise on the face verification), and these label errors will lead to poor learning dynamics.

4.4. Test time augmentation

[7] use average pooling at test time by evaluating the entire test utterance at once by changing the size of the *apool6* layer. Here, we experiment with different augmentation protocols for evaluating the performance at test time. We propose three methods:
(1) Baseline: variable average pooling as described in [7];
(2) Sample ten 3-second temporal crops from each test segment, and use the mean of the features;
(3) Sample ten 3-second temporal crops from each test segment, compute the distances between the every possible pair of crops ($10 \times 10 = 100$) from the two speech segments, and use the mean of the 100 distances. The method results in a marginal improvement in performance, as shown in Table 4.

4.5. Implementation Details

Input features. Spectrograms are computed from raw audio in a sliding window fashion using a hamming window of width 25ms and step 10ms, in exactly the same manner as [7]. This gives spectrograms of size 512 x 300 for 3 seconds of speech. Mean and variance normalisation is performed on every frequency bin of the spectrum.

Training. During training, we randomly sample 3-second segments from each utterance. Our implementation is based on the deep learning toolbox MatConvNet [38]. Each network is trained on three Titan X GPUs for 30 epochs or until the validation error stops decreasing, whichever is sooner, using a batch-size of 64. We use SGD with momentum (0.9), weight decay ($5E-4$) and a logarithmically decaying learning rate (initialised to 10^{-2} and decaying to 10^{-8}).

5. Results

Original `VoxCeleb1` test set. Table 4 provides the performance of our models on the original `VoxCeleb1` test set. As might be expected, performance improves with greater network depth, and also with more training data (`VoxCeleb2` vs `VoxCeleb1`). This also demonstrates that `VoxCeleb2` provides a suitable training regime for use on other datasets.

Models	**Trained on**	C_{det}^{min}	EER (%)
I-vectors + PLDA (1) [7]	`VoxCeleb1`	0.73	8.8
VGG-M (Softmax) [7]	`VoxCeleb1`	0.75	10.2
VGG-M (1) [7]	`VoxCeleb1`	0.71	7.8
VGG-M (1)	`VoxCeleb2`	0.609	5.94
ResNet-34 (1)	`VoxCeleb2`	0.543	5.04
ResNet-34 (2)	`VoxCeleb2`	0.553	5.11
ResNet-34 (3)	`VoxCeleb2`	0.549	4.83
ResNet-50 (1)	`VoxCeleb2`	0.449	4.19
ResNet-50 (2)	`VoxCeleb2`	0.454	4.43
ResNet-50 (3)	`VoxCeleb2`	**0.429**	**3.95**

Table 4: *Results for verification on the original* `VoxCeleb1` *test set (lower is better). The number in brackets refer to the test time augmentation methods described in Section 4.4.*

New `VoxCeleb1-E` test set – using the entire dataset. Popular speaker verification test sets in the wild [7, 8] are limited in the number of speakers. This yields the possible danger of optimising performance to overfit the small number of speakers in the test set, and results are not always indicative of good generalised performance. Hence we propose a new evaluation protocol consisting of 581,480 random pairs sampled from the entire `VoxCeleb1` dataset, covering 1,251 speakers, and set benchmark performance for this test set. The result is given in Table 5.

New `VoxCeleb1-H` test set – within the same nationality and gender. By using the whole of `VoxCeleb1` as a test set, we are able to test only on the pairs with same nationality and gender. We propose a new evaluation list consisting of 552,536 pairs sampled from the `VoxCeleb1` dataset, all of which are from the same nationality and gender. 18 nationality-gender combinations each with at least 5 individuals are used to generate this list, of which 'USA-Male' is the most common. The result is given in Table 5.

Models	**Tested on**	C_{det}^{min}	EER (%)
ResNet-50 (3)	`VoxCeleb1-E`	**0.524**	**4.42**
ResNet-50 (3)	`VoxCeleb1-H`	**0.673**	**7.33**

Table 5: *Results for verification on the extended* `VoxCeleb1` *test sets.*

6. Conclusion

In this paper, we have introduced new architectures and training strategies for the task of speaker verification, and demonstrated state-of-the-art performance on the `VoxCeleb1` dataset. Our learnt identity embeddings are compact ($512D$) and hence easy to store and useful for other tasks such as diarisation and retrieval. We have also introduced the `VoxCeleb2` dataset, which is several times larger than any speaker recognition dataset, and have re-purposed the `VoxCeleb1` dataset, so that the entire dataset of 1,251 speakers can be used as a test set for speaker verification. Choosing pairs from all speakers allows a better assessment of performance than from the 40 speakers of the original test set. We hope that this new test set will be adopted, alongside SITW, as a standard for the speech community to evaluate on.

Acknowledgements. Funding for this research is provided by the EPSRC Programme Grant Seebibyte EP/M013774/1.

7. References

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015.

[2] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC.*, 2015.

[4] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: a dataset for recognising faces across pose and age," *arXiv preprint arXiv:1710.08092*, 2017.

[5] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4873–4882.

[6] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 87–102.

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

[8] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *INTERSPEECH*, 2016.

[9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC.*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[11] 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, 2011.

[12] P. Matějka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Černocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4828–4831.

[13] S. Cumani, O. Plchot, and P. Laface, "Probabilistic linear discriminant analysis of i-vector posterior distributions," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7644–7648.

[14] 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.

[15] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[16] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[17] S. H. Ghalehjegh and R. C. Rose, "Deep bottleneck features for i-vector based text-independent speaker verification," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 555–560.

[18] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP, Calgary*, 2018.

[20] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, H. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vectors for on-device image matching," in *Asilomar*, 2011.

[21] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 402–406.

[22] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[23] D. van der Vloed, J. Bouten, and D. A. van Leeuwen, "NFI-FRITS: a forensic speaker recognition database and some first experiments," in *The Speaker and Language Recognition Workshop*, 2014.

[24] J. Hennebert, H. Melin, D. Petrovska, and D. Genoud, "POLYCOST: a telephone-speech database for speaker recognition," *Speech communication*, vol. 31, no. 2, pp. 265–270, 2000.

[25] J. B. Millar, J. P. Vonwiller, J. M. Harrington, and P. J. Dermody, "The Australian national database of spoken language," in *Proc. ICASSP*, vol. 1. IEEE, 1994, pp. I–97.

[26] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report*, vol. 93, 1993.

[27] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA speech recognition research database: specifications and status," in *Proc. DARPA Workshop on speech recognition*, 1986, pp. 93–99.

[28] C. S. Greenberg, "The NIST year 2012 speaker recognition evaluation plan," *NIST, Technical Report*, 2012.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. ECCV*. Springer, 2016, pp. 21–37.

[30] J. S. Chung and A. Zisserman, "Lip reading in profile," in *Proc. BMVC.*, 2017.

[31] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[32] J. S. Chung and A. Zisserman, "Learning to lip read words by watching videos," *CVIU*, 2018.

[33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, vol. 1. IEEE, 2005, pp. 539–546.

[34] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, vol. 2. IEEE, 2006, pp. 1735–1742.

[35] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[36] K.-K. Sung, "Learning and example selection for object and pattern detection," Ph.D. dissertation, 1996.

[37] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 4004–4012.

[38] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," *CoRR*, vol. abs/1412.4564, 2014.