Voxceleb: Large-scale speaker verification in the wild☆

Arsha Nagrani[1,*,a], Joon Son Chung[1,a,b], Weidi Xie[1,a], Andrew Zisserman[a]

[a] *Visual Geometry Group, Department of Engineering Science, University of Oxford, United Kingdom*
[b] *Naver Corporation, South Korea*

A R T I C L E   I N F O

A B S T R A C T

The objective of this work is speaker recognition under noisy and unconstrained conditions. We make two key contributions. First, we introduce a very large-scale *audio-visual* dataset collected from open source media using a *fully automated pipeline*. Most existing datasets for speaker identification contain samples obtained under quite constrained conditions, and usually require manual annotations, hence are limited in size. We propose a pipeline based on computer vision techniques to create the dataset from open-source media. Our pipeline involves obtaining videos from YouTube; performing active speaker verification using a two-stream synchronization Convolutional Neural Network (CNN), and confirming the identity of the speaker using CNN based facial recognition. We use this pipeline to curate VoxCeleb which contains contains over a million 'real-world' utterances from over 6000 speakers. This is several times larger than any publicly available speaker recognition dataset. Second, we develop and compare different CNN architectures with various aggregation methods and training loss functions that can effectively recognise identities from voice under various conditions. The models trained on our dataset surpass the performance of previous works by a significant margin.

1. Introduction

Speaker recognition under noisy and unconstrained conditions is an extremely challenging task. Applications of speaker recognition vary from authentication in high-security systems and forensic tests, to searching for persons in large corpora of speech data. All such tasks require high speaker recognition performance under 'real-world' conditions. This is extremely difficult due to both extrinsic and intrinsic variations; extrinsic variations include background chatter and music, laughter, reverberation, channel and microphone effects; while intrinsic variations are factors inherent to the speakers themselves such as age, accent, emotion, intonation and manner of speaking, amongst others (Stoll, 2011).

Deep Convolutional Neural Networks (henceforth, CNNs) have given rise to substantial improvements in speech recognition, computer vision and related fields due to their ability to deal with real-world, noisy datasets without the need for handcrafted features (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016a). One of the most important ingredients for the success of such methods, however, is the availability of large training datasets.

Unfortunately, large-scale public datasets in the field of speaker identification with unconstrained speech samples are lacking. While large-scale evaluations are held regularly by the National Institute of Standards in Technology (NIST), these datasets are not freely available to the research community. The only freely available dataset curated from multimedia is the Speakers in the

---

Wild (SITW) dataset (McLaren et al., 2016), which contains speech samples of 299 speakers across unconstrained or 'wild' conditions. This is a valuable dataset, as the speech samples have been manually annotated, however, scaling it further, for example to thousands of speakers across tens of thousands of utterances, would require the use of a service such as Amazon Mechanical Turk (AMT). In the computer vision community AMT like services have been used to produce very large-scale datasets, such as ImageNet (Russakovsky et al., 2015).

We make two contributions towards the goal of speaker recognition under noisy and unconstrained conditions. The first contribution is to propose a fully automated and scalable pipeline for creating a large-scale 'real-world' speaker identification dataset. Benefiting from the success of face recognition research in computer vision, our method circumvents the need for human annotation completely. We use this method to curate VoxCeleb[2], a large-scale dataset with over a million utterances for over seven thousand 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. The dataset is audio-visual, so is also useful for a number of other applications, for example, visual speech synthesis (Chung et al., 2017; Karras et al., 2017), speech separation (Afouras et al., 2018; Ephrat et al., 2018), cross-modal transfer from face to voice or vice versa (Nagrani et al., 2018b; 2018a), emotion recognition (Albanie et al., 2018) and training face recognition from video to complement existing face recognition datasets (Cao et al., 2018; Kemelmacher-Shlizerman et al., 2016; Guo et al., 2016). Since its official release in 2017, the dataset has already been downloaded over 3000 times.

The second contribution is a deep CNN based neural speaker verification system, named VGGVox, which is trained to map voice spectrograms to a compact embedding space. We then use the cosine distance between vectors in this embedding space to measure the similarity between speakers. Besides speaker recognition and verification, clustering and novel speaker discovery can be straightforwardly implemented using standard techniques, with our embeddings as features. In developing VGGVox we investigate current popular CNN architectures, *e.g.* variants of VGG-M (Chatfield et al., 2014) and ResNet (He et al., 2016a), different aggregation strategies, *e.g.* global average pooling, NetVLAD (Arandjelović et al., 2016), GhostVLAD (Zhong et al., 2018), and different loss functions for training the model, *e.g.* standard softmax classification, large-margin softmax and the contrastive loss. Our methods achieve state-of-the-art performance on the VoxCeleb1 speaker verification task, outperforming all other traditional methods and recent deep learning methods.

This paper consolidates three separate conference papers (Chung et al., 2018; Nagrani et al., 2017; Xie et al., 2019) into a single coherent document. In addition, we have added new results based on a new relation module, and added a more detailed comparison to previous work and the discussion of results.

2. Related works

Traditional methods. For a long time, speaker identification was dominated by Gaussian Mixture Models (GMMs) trained on low dimensional feature vectors (Reynolds et al., 2000; Reynolds and Rose, 1995). The state of the art in more recent times involves both the use of joint factor analysis (JFA) based methods which model speaker and channel subspaces separately (Kenny, 2005), and i-vectors which attempt to model both subspaces into a single compact, low-dimensional space (Dehak et al., 2011). These methods rely on a low dimensional representation of the audio input, such as Mel Frequency Cepstrum Coefficients (MFCCs). However, not only does the performance of MFCCs degrade rapidly in real-world noise (Yapanel et al., 2002; Hansen et al., 2001), but by focusing only on the overall spectral envelope of short frames, MFCCs may be lacking in speaker-discriminating features (such as pitch information) An in-depth review of these traditional methods is given in Hansen and Hasan (2015).

Deep learning methods. Deep neural networks (DNN) have been used successfully as feature extractors to learn discriminative embeddings in both computer vision and speech. Such methods (Variani et al., 2014; Lei et al., 2014; Ghalehjegh and Rose, 2015; Snyder et al., 2017; 2018) are often combined with classifiers, both being trained independently. While such fusion methods are highly effective, since they are not trained end to end 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. End-to-end deep learning based systems for speaker recognition usually follow a similar three-stage pipeline: (i) frame level feature extraction using a DNN; (ii) temporal aggregation of frame level features; and (iii) optimisation of a classification loss. In the following, we review the three components in turn.

The trunk DNN architecture used is often either a 2D CNN with convolutions in both the time and frequency domain (Nagrani et al., 2017; Chung et al., 2018; Cai et al., 2018c; 2018b; Hajibabaei and Dai, 2018; Bhattacharya et al., 2017), or a 1D CNN with convolutions applied only to the time domain (Snyder et al., 2017; Shon et al., 2018; Okabe et al., 2018; Snyder et al., 2018). A number of papers (Wan et al., 2018; Chowdhury et al., 2017) have also used LSTM-based front-end architectures, including the work by Heigold et al. (2016), which unlike our work focused on *text-dependant* speaker verification.

The output from the feature extractor is a variable length feature vector, dependant on the length of the input utterance. Average pooling layers have been used in Nagrani et al. (2017), Chung et al. (2018), and Wan et al. (2018) to aggregate frame-level feature vectors to obtain a fixed length utterance-level embedding. Snyder et al. (2017) introduces an extension of the method in which the standard deviation is used as well as the mean – this method is termed *statistical pooling*, and used by Shon et al.

---

angular softmax

(2018) and Snyder et al. (2018). Unlike these methods which ingest information from all frames with equal weighting, Bhatta-charya et al. (2017), and Chowdhury et al. (2017) have employed attention models to assign weight to the more discriminative frames. Okabe et al. (2018) combines the attention models and the statistical model to propose *attentive statistics pooling*. The final pooling strategy of interest is the Learnable Dictionary Encoding (LDE) proposed by Cai et al. (2018a,c). This method is closely based on the NetVLAD layer (Arandjelović et al., 2016) designed for image retrieval.

Typically, such systems are trained end-to-end for classification with a softmax loss (Okabe et al., 2018) or one of its variants, such as the angular softmax (Cai et al., 2018c). In some cases, the network is further trained for verification using the contrastive loss (Nagrani et al., 2017; Chung et al., 2018; Chen et al., 2011) or other metric learning losses such as the triplet loss (Li et al., 2017). Similarity metrics like the cosine similarity or PLDA are often adopted to generate a final pairwise score.

Datasets. Many existing datasets are obtained under controlled conditions, for example: forensic data intercepted by police officials (van der Vloed et al., 2014), data from telephone calls (Hennebert et al., 2000), speech recorded live in high quality environments such as acoustic laboratories (Millar et al., 1994; Garofolo et al., 1993), or speech recorded from mobile devices (McCool and Marcel, 2009; Woo et al., 2006). Morrison et al. (2015) consists of more natural speech but has been manually processed to remove extraneous noises and crosstalk. All the above datasets are also obtained from single-speaker environments, and are free from audience noise and overlapping speech.

Datasets obtained from multi-speaker environments include those from recorded meeting data (Janin et al., 2003; McCowan et al., 2005), or from audio broadcasts (Bell et al., 2015). These datasets usually contain audio samples under less controlled conditions. Some datasets contain artificial degradation in an attempt to mimic real-world noise, such as those developed using the TIMIT dataset (Garofolo et al., 1993): NTIMIT, (transmitting TIMIT recordings through a telephone handset) and CTIMIT, (passing TIMIT files through cellular telephone circuits).

Table 1 summarises existing speaker identification datasets. Besides lacking real-world conditions, to the best of our knowledge, most of these datasets have been collected with great manual effort, other than (Bell et al., 2015) which was obtained by mapping subtitles and transcripts to broadcast data.

3. The VoxCeleb dataset

We released the dataset in two stages, as VoxCeleb1 and VoxCeleb2. VoxCeleb1 contains over 100,000 utterances for 1251 celebrities, while VoxCeleb2 contains over 1 million utterances for over 6000 celebrities extracted from videos uploaded to You-Tube. The datasets are fairly gender balanced, (VoxCeleb1 − 55% male, VoxCeleb2 − 61% 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 indoor studios, speeches given to large audiences, excerpts from professionally shot multimedia, and even crude videos shot on hand-held devices. Crucially, all are degraded with real-world noise, consisting of background chatter, laughter, overlapping speech, room acoustics, and there is a range in the quality of recording equipment and channel noise. 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 2 gives the general statistics, and Fig. 1 shows examples of cropped faces as well as utterance length, gender and nationality distributions.

Both datasets contain development and test sets, with disjoint speakers. The development set of VoxCeleb2 has no overlap with the identities in the VoxCeleb1 or SITW datasets. Since we have created a number of evaluation benchmarks using the Vox-Celeb1 dataset for testing (Section 6.1), we encourage others to use the *development* set of VoxCeleb2 *only* to train models for the speaker recognition task (Sections 5−7) so that they can evaluate their methods fairly on VoxCeleb1. The VoxCeleb2 *test* set should prove useful for other applications of audio-visual learning for which the dataset might be used. The statistics for all the

import

Table 1
Comparison of existing speaker identification datasets. **Cond.:** Acoustic conditions; **Utter.:** Approximate number of utterances. †And its derivatives. ‡Number of telephone calls. * varies by year. ** Only available to participants of the challenge. This dataset was mainly used for speech recognition (ASR).

Name	Cond.	Free	# of Speakers	# of Utter.
ELSDSR (Feng and Hansen, 2005)	Clean Speech	✓	22	198
MIT Mobile (Woo et al., 2006)	Mobile Devices	−	88	7884
SWB (Godfrey et al., 1992)	Telephony	−	3114	33,039
POLYCOST (Hennebert et al., 2000)	Telephony	−	133	1285‡
ICSI Meeting Corpus (Janin et al., 2003)	Meetings	−	53	922
Forensic Comparison (Morrison et al., 2015)	Telephony	✓	552	1264
ANDOSL (Millar et al., 1994)	Clean speech	−	204	33,900
TIMIT (Fisher et al., 1986)†	Clean speech	−	630	6300
MGB Challenge Dataset (Bell et al., 2015)	Broadcast Data	**	Unknown	1600 hs
SITW (McLaren et al., 2016)	Multi-media	✓	299	2800
NIST SRE Greenberg (2012)	Clean speech	−	2000+	*
VoxCeleb1	Multi-media	✓	**1251**	**153,516**
VoxCeleb2	Multi-media	✓	**6112**	**1,128,246**

Table 2
Dataset statistics for both VoxCeleb1 and VoxCeleb2. Note VoxCeleb2 is more than 5 times larger than VoxCeleb1.

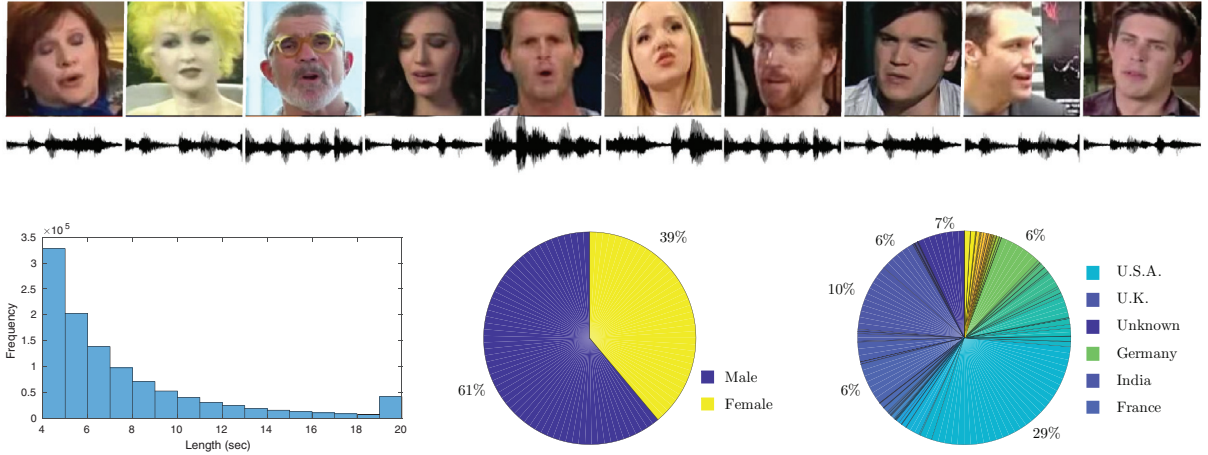| Dataset	VoxCeleb1	VoxCeleb2
# of speakers	1251	6112
# of male speakers	690	3761
# of videos	22,496	150,480
# of hours	352	2442
# of utterances	153,516	1,128,246
Avg # of videos per speaker	18	25
Avg # of utterances per speaker	116	185
Avg length of utterances (s)	8.2	7.8

Fig. 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.

dev/test splits are given in Table 3. For clarity, we also provide a summary of the possible overlap between the developement and test sets of VoxCeleb1, VoxCeleb2 and SITW in Table 4. This is useful for researchers wishing to train on one of these datasets and test on another.

4. Dataset collection pipeline

This section describes our multi-stage approach for collecting a large speaker recognition dataset, starting from YouTube videos. Our pipeline involves obtaining videos from YouTube; performing active speaker verification using a two-stream

Table 3
Development and test set splits for VoxCeleb1 and VoxCeleb2.

Dataset	V1 Dev	V1 Test	V1 All	V2 Dev	V2 Test	V2 All
# of speakers	1211	40	1251	5994	118	6112
# of videos	21,819	677	22,496	145,569	4911	150,480
# of utterances	148,642	4874	153,516	1,092,009	36,237	1,128,246

Table 4
Overlap between development and test sets for VoxCeleb1, VoxCeleb2 and SITW. N refers to there definitely being no overlap, Y refers to the possibility of overlap between the sets.

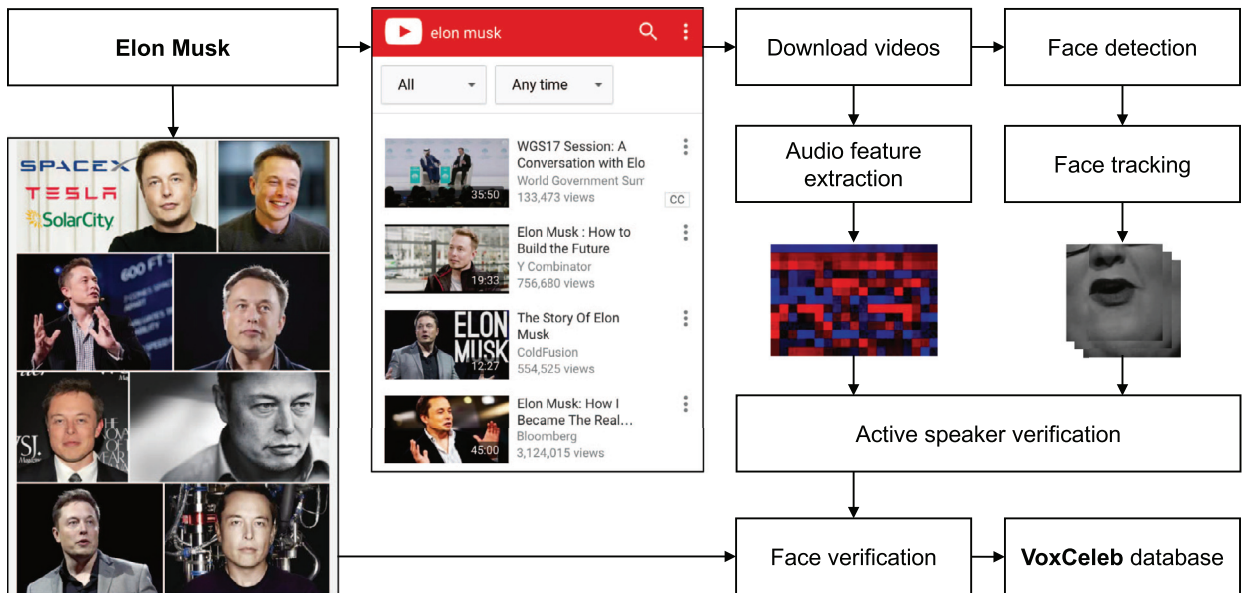|	Vox1 Train	Vox1 Test	Vox2 Train	Vox2 Test	SITW
Vox1 Train	Y	N	N	Y	Y
Vox1 Test	N	Y	N	Y	Y
Vox2 Train	N	N	Y	N	N
Vox2 Test	Y	Y	N	Y	Y
SITW	Y	Y	N	Y	Y

Fig. 2. The multi-stage automatic pipeline used to create the VoxCeleb dataset automatically from YouTube videos. Our pipeline involves obtaining videos from YouTube; performing active speaker verification using a two-stream synchronization Convolutional Neural Network (CNN), and confirming the identity of the speaker using CNN based facial recognition.

synchronization Convolutional Neural Network (CNN), and confirming the identity of the speaker using CNN based facial recognition. Using this fully automated pipeline, we have obtained over a million utterances for thousands of different speakers. The overall pipeline is the same for both VoxCeleb1 and VoxCeleb2, but the methods used in key stages differ since we selected the state-of-the-art face recognition systems at the time of dataset curation. The pipeline is summarised in Fig. 2, and key stages are discussed in the following subsections:

4.1. Candidate list of speakers

The first stage is to obtain a list of speakers.

VoxCeleb1. We start from the list of people that appear in the VGGFace1 dataset (Parkhi et al., 2015), which is based on an intersection of the most searched names in the Freebase knowledge graph, and the Internet Movie Data Base (IMDB). This list contains 2622 identities, ranging from actors and sportspeople to entrepreneurs, of which approximately half are male and the other half female.

VoxCeleb2. The list of candidate names are drawn from the VGGFace2 dataset (Cao et al., 2018), which has greater ethnic diversity compared to VGGFace1. This list contains over 9000 identities, ranging from actors and sportspeople to politicians. There are a number of overlapping identities between VGGFace1 and VGGFace2 – these are excluded from the development set of VoxCeleb2, so that any models trained on VoxCeleb2 can be tested on VoxCeleb1.

4.2. Downloading videos from YouTube

The top 50 or 100 videos for each of the speakers are automatically downloaded using YouTube search for VoxCeleb1 and VoxCeleb2, respectively. The word 'interview' is appended to the name of the speaker in search queries to increase the likelihood that the videos contain an instance of the speaker speaking, and to filter out sports or music videos. No other filtering is done at this stage.

4.3. Face tracking

VoxCeleb1. The HOG-based face detector (King, 2009) is used to detect the faces in every frame of the video. Facial landmark positions are detected for each face detection using the regression tree based method of Kazemi and Sullivan (2014).

VoxCeleb2. The CNN face detector based on the Single Shot MultiBox Detector (SSD) (Liu et al., 2016) is used to detect face appearances on every frame of the video. This detector is a distinct improvement over (King, 2009), allowing the detection of faces in profile and extreme poses.

For both datasets, the shot boundaries are detected by comparing colour histograms across consecutive frames. Within each detected shot, face detections are grouped together into face tracks using a position-based tracker. This stage is closely related to

the tracking pipeline of Chung and Zisserman (2016a) and Everingham et al. (2009), but optimised to reduce run-time given the very large number of videos to process.

4.4. Active speaker verification

The goal of this stage is to determine the audio-video synchronisation between mouth motion and speech in a video in order to determine which (if any) visible face is the speaker. This is done by using 'SyncNet', a two-stream CNN described in Chung and Zisserman (2016b, 2018) which estimates the correlation between the audio track and the mouth motion of the video. For VoxCeleb2, the SyncNet model is replaced with a multi-view variant (Chung and Zisserman, 2017), so that talking faces can be detected even when the face is off-frontal. This method is able to reject the clips that contain dubbing or voice-over.

4.5. Face verification

Active speaker face tracks are then classified into whether they are of the speaker or not using the VGGFace and VGGFace2 CNNs for VoxCeleb1 and VoxCeleb2, respectively. Verification is done by directly comparing the cosine similarity of the face embedding from the pretrained networks – the face classification networks have been trained on images of the same set of speakers (the VGGFace CNN is trained on the VGGFace image dataset, and VoxCeleb1 starts from the same list of speakers, similarly for VGGFace2).

4.6. 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 Nagrani et al. (2017) 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 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.

4.7. Manual filtering

Since VoxCeleb1 is intended to be used as a test set for speaker verification, the data is checked manually for any errors. This is done using a simple web-based tool that shows all video segments for each identity. In order to highlight the segments which are more likely to contain errors, face and voice embeddings are generated from SphereFace Liu et al. (2017) and our own model trained on VoxCeleb2, respectively, and those with lower confidence are highlighted with a different colour. By running this check, we discovered around 300 label errors, which account for around 0.2% of the VoxCeleb1 data.

4.8. 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 36 nationalities for VoxCeleb1 and 145 for VoxCeleb2. The VoxCeleb2 is a far more ethnically diverse dataset, with a smaller percentage of U.S. speakers (29% in VoxCeleb2 compared to 64% in VoxCeleb1).

4.9. Discussion

In order to ensure that our system is extremely confident that a person is speaking (Section 4.4), and that they have been correctly identified (Section 4.5) without any manual interference, we set very conservative thresholds in order to minimise the number of false positives. This conservative threshold allows us to operate in a high precision low recall regime. The large number of videos downloaded initially allows us to discard many, and only keep the ones with extremely high confidence. Precision-recall curves for both tasks on their respective benchmark datasets (Parkhi et al., 2015; Chakravarty and Tuytelaars, 2016) are shown in Fig. 3, and the values at the operating point are given in Table 5. Employing these thresholds ensures that although we discard a lot of the downloaded videos, we can be reasonably certain that the dataset has few labelling errors. 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 videos are discarded.

This ensures an automatic pipeline that can be scaled up to any number of speakers and utterances (if available) as required.
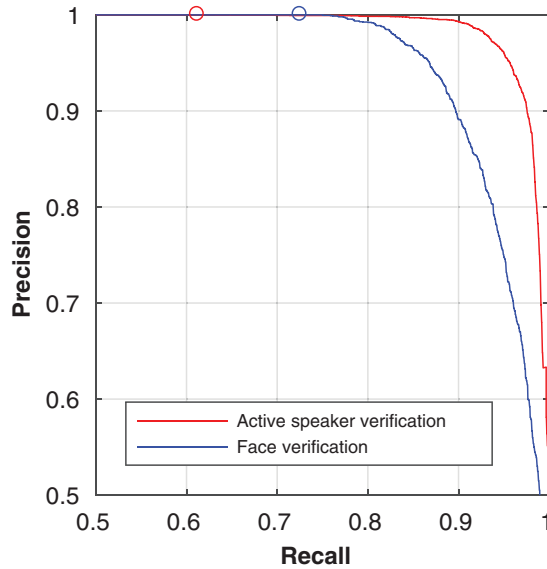
Fig. 3. Precision-recall curves for the active speaker verification (using a 25-frame window) and the face verification steps, tested on standard benchmark datasets (Parkhi et al., 2015; Chakravarty and Tuytelaars, 2016). Operating points are shown in circles.

Table 5
Precision-recall values at the chosen operating points for VoxCeleb1.

Task	Dataset	Precision	Recall
Active speaker verification	Chakravarty and Tuytelaars (2016)	1.000	0.613
Face verification	Parkhi et al. (2015)	1.000	0.726

5. VGGVox

In this section we describe our neural embedding system, called VGGVox. Our aim is to move from techniques that require traditional hand-crafted features, to a CNN architecture that can train end-to-end for the task of speaker recognition. 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, and the features are aggregated to obtain utterance-level speaker embeddings. The entire model is then trained end-to-end.

We use 2D CNNs as feature extractors and treat 2D spectrograms as single-channel images. It is perhaps unnatural to treat spectrograms in this manner where the same convolution is used at every point since, unlike in a visual image where an object may appear at any location, a pattern can appear at any point on the time axis but we would not expect patterns to also be frequency independent. However, deep networks can potentially learn frequency-specific filters if they are needed for solving a downstream task; for instance, some filters can only fire on specific patterns existing in the low frequency region, whilst fully connected layers can be position dependent. Therefore, even if a 2D CNN uses shared filters on the spectrogram, the model has the capability to divide the filters into low/high frequency groups.

We experiment with different trunk architectures, aggregation strategies as well as training losses. We describe the trunk and aggregation architectures here, and the losses in Section 6.2.

5.1. Input features

We use short-term magnitude spectrograms as input to our deep CNN architecture. Mean and variance normalisation is performed on every frequency bin of the spectrum. No other speech-specific preprocessing (e.g., silence removal, voice activity detection, or removal of unvoiced speech) is used. Precise implementation details are provided in Section 6.3.

5.2. Trunk architecture

We experiment with both VGG (Chatfield et al., 2014) and ResNet style CNN architectures.

VGG-M: The baseline trunk architecture is the CNN introduced in Nagrani et al. (2017). This architecture is a modification of the VGG-M (Chatfield et al., 2014) CNN, known for high efficiency and good performance on image classification. The modification concerns the addition of an aggregation layer, and is described below. The complete CNN architecture is specified in Table 6.

Table 6
VGG style architecture. The data size on the right is the *output* data size for each layer. Here we assume input spectrograms of size 512 × 300, and up to *fc6* the sizes have been calculated for an input with a temporal dimension of 300, but the network is able to accept inputs of variable lengths. Note that the first layer also has zero padding.

Layer	Support	Filt dim.	# filts.	Stride	Data size
conv1	7 × 7	1	96	2 × 2	254 × 148
mpool1	3 × 3	–	–	2 × 2	126 × 73
conv2	5 × 5	96	256	2 × 2	62 × 36
mpool2	3 × 3	–	–	2 × 2	30 × 17
conv3	3 × 3	256	384	1 × 1	30 × 17
conv4	3 × 3	384	256	1 × 1	30 × 17
conv5	3 × 3	256	256	1 × 1	30 × 17
mpool5	5 × 3	–	–	3 × 2	9 × 8
fc6	9 × 1	256	4096	1 × 1	1 × 8
apool6	1 × n	–	–	1 × 1	1 × 1
fc7	1 × 1	4096	1024	1 × 1	1 × 1
fc8	1 × 1	1024	1251	1 × 1	1 × 1

ResNets: The residual-network (ResNet) architecture (He et al., 2016b) 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. In this paper, we experiment with three variants of ResNets, e.g., ResNet-34, ResNet-50 and a Thin-ResNet which contains fewer parameters. We modify the layers to adapt to the spectrogram input. The architectures are specified in Table 7.

5.3. Aggregation strategies

Features produced by the trunk CNN architecture are then aggregated in time to produce a single fixed length representation for each audio input. We experiment with two aggregation strategies: simple non-trainable average pooling, as well as a trainable aggregation layer based on the NetVLAD layer. Here we provide a brief overview of both the average pooling aggregation layer, and also the NetVLAD (for full details please refer to Arandjelović et al. (2016)).

Average pooling aggregation. 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 aggregation layer – global average pooling along the temporal axis. 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 (for our given input size this reduction is fivefold, i.e., from 319M in VGG-M to 67M in our network) which helps avoid overfitting.

Table 7
Modified ResNet34, ResNet50 and Thin-ResNet architectures with average pool layer at the end. Batch normalisation is used before the rectified linear unit (ReLU) activations. Each row specifies the number of convolutional filters and their sizes as **size × size, # filters**. Square brackets indicate blocks over which there are residual connections.

Layer name	ResNet34	ResNet50	Thin-ResNet
conv1	7 × 7, 64, stride 2	7 × 7, 64, stride 2	7 × 7, 64, stride 1
pool1	3 × 3, max pool stride 2	3 × 3, max pool stride 2	3 × 3, max pool stride 2
conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 48 \\ 3 \times 3, 48 \\ 1 \times 1, 96 \end{bmatrix} \times 2$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 96 \\ 3 \times 3, 96 \\ 1 \times 1, 128 \end{bmatrix} \times 3$
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 512 \end{bmatrix} \times 3$
fc1	9 × 1, 512, stride 1	9 × 1, 2048, stride 1	7 × 1, 512, stride 1
pool_time	1 × N, avg pool stride 1	1 × N, avg pool stride 1	3 × N, max pool stride 2

NetVLAD aggregation The CNN trunk architecture maps the input spectrogram to frame-level descriptors, as described in the Thin-ResNet shown in Table 7, the output feature is downsampled by a factor of 32. The NetVLAD layer then takes dense descriptors as input and produces a single $K \times D$ matrix V, where K refers to the number of chosen cluster, and D refers to the dimensionality of each cluster. Concretely, the matrix of descriptors V is computed using the following equation:

$$V(k,j) = \sum_{t=1}^{T/32} \frac{e^{w_k^T x_t + b_k}}{\sum_{k'=1}^{K} e^{w_{k'}^T x_t + b_{k'}}} \left(x_t(j) - c_k(j) \right) \tag{1}$$

where $\{w_k\}$, $\{b_k\}$ and $\{c_k\}$ are trainable parameters, with $k \in [1, 2, \ldots, K]$. The first term corresponds to the soft-assignment weight of the input vector x_i for cluster k, while the second term computes the residual between the vector and the cluster centre. Each row in V, i.e., the residual from each cluster is then L2 normalized. The final output is then obtained by flattening this matrix into a long vector, i.e. row vectors are concatenated. To keep computational and memory requirements low, dimensionality reduction is performed via a Fully Connected (FC) layer, where we pick the output dimensionality to be 512. We also experiment with the recently proposed **GhostVLAD** Zhong et al. (2018) layer, where some of the clusters are not included in the final concatenation, and so do not contribute to the final representation, these are referred to as 'ghost clusters' (we used *two* in our implementation). Therefore, while aggregating the frame-level features, the contribution of the noisy and undesirable sections of a speech segment to normal VLAD clusters is effectively down-weighted, as most of their weights have been assigned to the 'ghost cluster'. For further details, please refer to Zhong et al. (2018).

6. Experiments

This section describes our experimental setup for speaker verification, loss functions, baselines, and implementation details. Along with releasing the VoxCeleb dataset, we also release a number of different evaluation benchmarks for testing speaker verification. These have been used extensively by the speech community to compare methods. In particular, we provide both easy pairs and hard pairs for testing; for the hard pairs, speakers with the same nationality and gender are chosen which makes distinguishing between them more challenging. This is described in more detail in the next section.

6.1. Evaluation splits and metrics

The methods are evaluated on a number of different test sets. These are described below and summarised in Table 8. All test set lists can be found on the VoxCeleb website[3].

OriginalVoxCeleb1test set. The original verification test set from VoxCeleb1 consists of 40 speakers. All speakers with names starting with 'E' are reserved for testing, since this gives a good balance of male and female speakers.

Extended VoxCeleb1-E test set − using the entire dataset. Since the above test set is limited in the number of speakers, there is a danger that models achieving high performance on this test set might not generalise to other sets of speakers. Hence we also propose a larger test set of 581,480 random pairs sampled from the entire VoxCeleb1 dataset, covering 1251 speakers.

Hard VoxCeleb1-H test set − within the same nationality and gender. This is a 'hard' evaluation set consisting of 552,536 pairs with the same nationality and gender, sampled from the entire VoxCeleb1 dataset. There are 18 nationality-gender combinations each with at least 5 individuals, of which 'USA-Male' is the most common.

Evaluation metric. We evaluate the models with Equal Error Rate (EER) and the minimum detection cost function (minDCF). EER measures the value at which the false-reject (miss) rate equals the false-accept (false-alarm) rate, and minDCF is defined as a weighted sum of false-reject and false-accept error probabilities. These are common metrics used by existing datasets and challenges, such as NIST SRE12 Greenberg (2012) and SITW McLaren et al. (2016).

6.2. Training loss

We experiment with a number of different training losses.

Softmax + Contrastive Loss. We employ a contrastive loss (Chopra et al., 2005; Hadsell et al., 2006) on paired embeddings, which seeks to minimise the distance between the embeddings of positive pairs and penalises the negative pair distances for

Table 8
VoxCeleb test sets.

Dataset	# of speakers	# of utterances	# of pairs
VoxCeleb1	40	4715	37,720
VoxCeleb1 (cleaned)	40	4708	37,611
VoxCeleb1-E	1251	145,375	581,480
VoxCeleb1-E (cleaned)	1251	145,160	579,818
VoxCeleb1-H	1190	138,137	552,536
VoxCeleb1-H (cleaned)	1190	137,924	550,894

---

[3] http://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox2.html

being smaller than a margin parameter α. Pair-wise losses such as the contrastive loss are notoriously difficult to train (Hermans et al., 2017), and hence to avoid suboptimal local minima early on in training, we proceed in two stages: first, pre-training for identification using a softmax loss, then, second, fine-tuning with the contrastive loss (described in more detail below).

Additive Margin Softmax. Besides the standard softmax loss, we also experiment with the additive margin softmax (AM-Softmax) classification loss (Wang et al., 2018) during training. This loss is designed explicitly for improving verification performance by introducing a margin in the angular space, meaning that we do not need to train with the contrastive loss after. The loss is given by the following equation:

$$L_i = -log \frac{e^{s(\cos\theta_{y_i}-m)}}{e^{s(\cos\theta_{y_i}-m)}+\sum_{j\neq y_i} e^{s\cos(\theta_j)}} \tag{2}$$

where L_i refers to cost of assigning the sample to the correct class, $\theta_y = \arccos(w^T x)$ refers to the angle between sample features (x) and the decision hyperplane (w), as both vectors have been L2 normalised. The goal is therefore to minimise this angle by making $\cos(\theta_{y_i}) - m$ as large as possible, where m refers to the angular margin. The hyper-parameter s controls the "temperature" of the softmax loss, producing higher gradients to the well-separated samples (and further shrinking the intra-class variance). We used the default values $m = 0.4$ and $s = 30$ (Wang et al., 2018).

Relation Loss. In this work, as another contribution, we introduce a novel relation module as a scoring mechanism. It is similar to a contrastive loss function, but uses a simple binary classifier rather than Euclidean distance. The relation module is shown in Fig. 4. It is inspired by the relation networks and their use in face comparisons (Santoro et al., 2017; Xie et al., 2018).

A Siamese network is constructed from two standard classification models, *i.e.* two branches share the same network and parameters (ThinResNet is fixed until conv4_x, refer to Table 7) that have been pretrained for speaker classification based on standard softmax, and the small relation module is then trained to distinguish if two voice samples are from same identity or not (binary classifier, implemented as a softmax with two classes). A separate NetVLAD/GhostVLAD aggregator is incorporated for the classification and relation network paths. As most of the feature extractor are fixed, the relation module only costs a very limited additional computation. During inference the output scores of a voice pair is computed as the average of the cosine similarity (between feature embeddings) and the classification score (from the small relation module).

6.3. Implementation details and training

During training, we randomly sample segments from each utterance. For the VGG based model, we use 3-s long segments with a 1024 FFT giving us spectrograms of size 512×300, and for the Thin-Resnet model we use 512 point FFTs giving us spectograms of size 257×250 (frequency × temporal). Earlier models (ResNet34 and ResNet50) are trained using the deep learning toolbox MatConvNet (Vedaldi and Lenc, 2014), and the latest models (Thin-ResNet) are in Keras (tensorflow).[4] The models and training code are publically available.[5] The model is trained using a fixed size spectrogram corresponding to a 2.5 s interval. All audio is first converted to single-channel, 16-bit streams at a 16kHz sampling rate for consistency. Spectrograms are then generated in a sliding window fashion using a hamming window of width 25ms and step 10ms. We normalise the spectrogram by subtracting the mean and dividing by the standard deviation of all frequency components in a single time step. No voice activity detection (VAD), or automatic silence removal is applied. We use the Adam optimizer with an initial learning rate of 0.001, and decrease the learning rate by a factor of 10 after every 36 epochs until convergence.
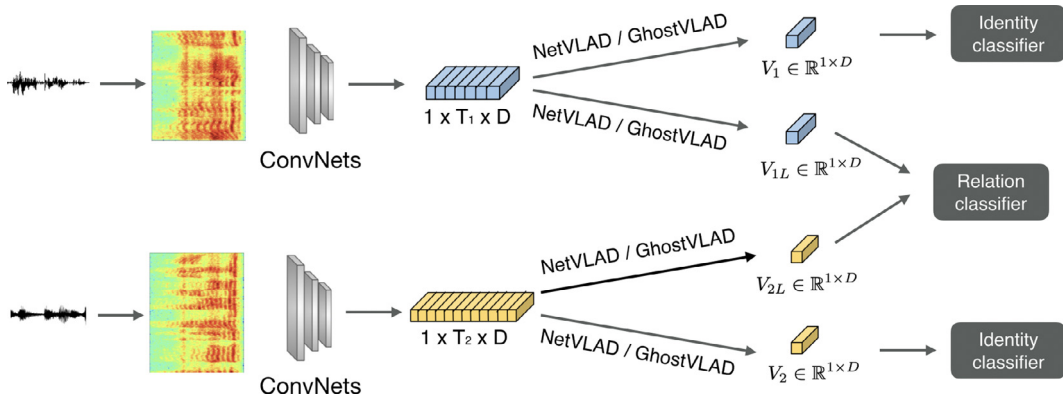

Fig. 4. Relation module. The relation module is added to a Siamese network in training and inference. During training, two sets of aggregation modules are used: V_1 and V_2 are computed from a shared NetVLAD/GhostVLAD and trained for identity classification; V_{1L} and V_{2L} are computed from a shared NetVLAD/GhostVLAD and used to train the relation classifier for identity matching.

---

[4] For our earlier spectrogram generation code, we discard the DC component, i.e., a 1024 pt FFT gives us spectrograms with 1024/2 frequency channels, but in later models we added the DC component to the spectrogram, hence for a 512 pt FFT we get spectrograms with $512/2+1$ frequency channels.

[5] http://www.robots.ox.ac.uk/~vgg/research/speakerID/ Xie et al. (2019).

Pre-training for contrastive loss. 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.

We take the model pre-trained on the identification task, and replace the classification layer with a fully connected layer of output dimension 512. This network is then trained with the contrastive loss.

Mining hard examples. 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 (Sung, 1996; Hermans et al., 2017; Song et al., 2016) 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. While training the relation module, a similar strategy is applied, we pre-compute the feature embeddings for all the voice samples in the entire VoxCeleb2 dataset. In addition to negative pairs, we mine both hard positive and negative pairs for training relation modules.

6.4. Non deep learning based baselines

For the sake of comparison, we also implement some traditional non-CNN methods and train them on the VoxCeleb1 dev set.

GMM-UBM. The GMM-UBM system uses MFCCs of dimension 13 as input. Cepstral mean and variance normalisation (CMVN) is applied on the features. Using the conventional GMM-UBM framework, a single speaker-independent universal background model (UBM) of 1024 mixture components is trained for 10 iterations from the training data.

I-vectors/PLDA. Gender independent i-vector extractors (Dehak et al., 2011) are trained on the VoxCeleb1 dataset to produce 400-dimensional i-vectors. Probabilistic LDA (PLDA) (Ioffe, 2006) is then used to reduce the dimension of the i-vectors to 200.

Inference. For identification, a one-vs-rest binary SVM classifier is trained for each speaker m ($m \in 1 \ldots K$). All feature inputs to the SVM are L2 normalised and a held out validation set is used to determine the C parameter (determines trade off between maximising the margin and penalising training errors). Classification during test time is done by choosing the speaker corresponding to the highest SVM score. The PLDA scoring function (Ioffe, 2006) is used for verification.

7. Results

In this section, we show all the evaluation results on three publicly available test sets created from VoxCeleb1, *i.e.* VoxCeleb1 test-set, VoxCeleb1-E, VoxCeleb1-H. Discussions of our main observations from these experiments are included, *e.g.*, benefits from the end-to-end trained CNN, size of training data, network architecture, different loss functions, and choice of aggregation strategy. We also compare performance of the CNN architectures to a number of other deep learning methods and more traditional state of the art methods.

7.1. Results on VoxCeleb1

7.1.1. Comparison to non-CNN methods
Comparing with the baseline methods that are based on traditional methods, *e.g.*, GMM-UBM, I-vectors+PLDA, achieving 15.0% EER and 8.8% EER on the standard VoxCeleb1 testing set, respectively, most of the Neural Networks (NN) based methods have shown clear advantages, for instance, one of our earliest VGG-M models (Nagrani et al., 2017) trained with Softmax and Contrastive has outperformed the traditional methods (obtaining 7.8% EER).

7.1.2. Size of training data
Deep neural networks are well-known for their capability to process large amount of data, in this section, we focus on exploring the benefits from a large dataset. In our experiments, we keep all settings unchanged, and only varying the datasets from VoxCeleb1 to VoxCeleb2, e.g. VGG-M model trained with Softmax+Contrastive (Nagrani et al., 2017; Chung et al., 2018). When testing on the standard VoxCeleb1 test set, larger training set (VoxCeleb2) leads to better performance (5.94% EER vs. 7.8% EER). This is due to the fact that we expect a larger dataset to naturally provide more diversity and variation, and therefore lead to better generalization.

7.1.3. Effect of CNN architecture
Following the continuous development of new architectures in computer vision, we also experiment with different trunk architectures, ranging from VGG to ResNet. In this comparison, we fix all the experimental settings and only vary the network architecture, i.e., we compare three models (VGG-M, ResNet-34, ResNet-50) trained with temporal average pooling (TAP), and Softmax+Contrastive loss on VoxCeleb2. Evaluation is done on the standard VoxCeleb1 test set without any test-time augmentation. Similar to the observations found in computer vision research, deeper networks lead to better generalization, therefore, ResNet-50 (4.19% EER) outperforms the ResNet-34 (5.04% EER) and VGG-M (5.94% EER), despite the fact that the VGG-M model uses a higher dimensional embedding (1024D).

Table 9
Comparison of our different models for verification on the original VoxCeleb1 test set (Nagrani et al., 2017) and the extended and hard test sets (VoxCeleb-E and VoxCeleb-H) (Chung et al., 2018). TAP: Temporal Average Pooling. TTA: Test Time Augmentation. † Cleaned up versions of the test lists have been released publically. We encourage other researchers to evaluate on these lists.

	Front-end model	Loss	Dims	Aggregation	Training set	EER (%)
VoxCeleb1 test set						
INTERSPEECH17 (Nagrani et al., 2017)	GMM-UBM	–	–	–	VoxCeleb1	15.0
INTERSPEECH17 (Nagrani et al., 2017)	I-vectors+PLDA	–	–	–	VoxCeleb1	8.8
INTERSPEECH17 (Nagrani et al., 2017)	VGG-M	Softmax	1024	TAP	VoxCeleb1	10.2
INTERSPEECH17 (Nagrani et al., 2017)	VGG-M	Softmax+Contrastive	1024	TAP	VoxCeleb1	7.8
INTERSPEECH18 (Chung et al., 2018)	VGG-M	Softmax+Contrastive	1024	TAP	VoxCeleb2	5.94
INTERSPEECH18 (Chung et al., 2018)	ResNet-34	Softmax+Contrastive	512	TAP	VoxCeleb2	5.04
INTERSPEECH18 (Chung et al., 2018) TTA-2	ResNet-34	Softmax+Contrastive	512	TAP	VoxCeleb2	5.11
INTERSPEECH18 (Chung et al., 2018) TTA-3	ResNet-34	Softmax+Contrastive	512	TAP	VoxCeleb2	4.83
INTERSPEECH18 (Chung et al., 2018)	ResNet-50	Softmax+Contrastive	512	TAP	VoxCeleb2	4.19
INTERSPEECH18 (Chung et al., 2018) TTA-2	ResNet-50	Softmax+Contrastive	512	TAP	VoxCeleb2	4.43
INTERSPEECH18 (Chung et al., 2018) TTA-3	ResNet-50	Softmax+Contrastive	512	TAP	VoxCeleb2	3.95
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	TAP	VoxCeleb2	10.48
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	NetVLAD	VoxCeleb2	3.57
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	AM-Softmax	512	NetVLAD	VoxCeleb2	3.32
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**3.22**
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	AM-Softmax	512	GhostVLAD	VoxCeleb2	3.23
ICASSP19 (**cleaned** †) (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	3.24
Ours + Relation Module	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**2.87**
VoxCeleb1-E						
INTERSPEECH18 (Chung et al., 2018)	ResNet-50	Softmax+Contrastive	512	TAP	VoxCeleb2	4.42
ICASSP19	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**3.24**
ICASSP19 (cleaned †)	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**3.13**
Ours + Relation Module †	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**2.95**
VoxCeleb1-H						
INTERSPEECH18 (Chung et al., 2018)	ResNet-50	Softmax+Contrastive	512	TAP	VoxCeleb2	7.33
ICASSP19 (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**5.17**
ICASSP19 (**cleaned** †) (Xie et al., 2019)	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**5.06**
Ours + Relation Module †	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**4.93**

7.1.4. Aggregation strategy and training loss

We next explore different aggregation methods and loss functions in this section. Once again, other experimental settings are fixed, for instance, we train the same Thin-ResNet on VoxCeleb2 and only vary the aggregation strategy (TAP, NetVLAD, GhostVLAD) and training loss (Softmax vs. AMSoftmax). As shown in Table 9, the Thin-ResNet trained with standard softmax loss and NetVLAD aggregation layer outperforms the previous model Chung et al. (2018) by a significant margin (EER of 3.57% vs 4.19%). The fact that the Thin-ResNet is actually shallower than the ResNet-50 (Table 7), and contain fewer number of parameters, further illustrates the benefits of the NetVLAD aggregation layer. By replacing the standard softmax with the additive margin softmax (AM-Softmax), a further performance gain is achieved (3.32% EER). The GhostVLAD layer, which excludes irrelevant information from the aggregation, additionally makes a modest contribution to performance (3.22% EER).

On the challenging VoxCeleb1-H test set, we outperform the previous best architecture Chung et al. (2018) (EER of 5.17% vs. 7.33%), which is by a larger margin than on the original VoxCeleb1 test set. We note that training a softmax loss based on features from temporal average pooling (TAP) yields extremely poor results (EER of 10.48%). We conjecture that the features trained using a softmax loss are typically good at separating different speakers), but not good at reducing the intra-class variation (i.e., making features of the same speaker compact). Therefore, contrastive loss with online hard sample mining leads to a significant performance boost, as demonstrated in Chung et al. (2018) for TAP.

7.1.5. Test time augmentation

Here, we experiment with different augmentation protocols for evaluating the performance at test time. We propose three methods:

Baseline: here we use variable average pooling where we evaluate the entire test utterance at once, by changing the size of an average pooling layer during test time according to the length of the test sample;

(TTA-2) Here we sample ten 3-s temporal crops from each test segment, and take the mean of the final embeddings;

(TTA-3) Here we sample ten 3-s 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. This final method results in a marginal improvement in performance, as shown in Table 9.

Table 10

Comparison of our best performing model to the state-of-the-art on the VoxCeleb1 original test set. TAP: Temporal Average Pooling. SAP: Self-attentive Pooling Layer Cai et al. (2018c). SP: Statistical Pooling. TTA: Test Time Augmentation. † We calculate minDCF using the standard parameters used in the NIST SRE 18[6].

	Front-end model	Loss	Dims	Aggregation	Training set	EER (%)	minDCF(0.01)†
				VoxCeleb1 test set			
Cai et al. (2018c)	ResNet-34	A-Softmax+PLDA	128	TAP	VoxCeleb1	4.46	–
Cai et al. (2018c)	ResNet-34	A-Softmax+PLDA	128	SAP	VoxCeleb1	4.40	–
Cai et al. (2018c)	ResNet-34	A-Softmax+PLDA	128	LDE	VoxCeleb1	4.48	–
Okabe et al. (2018)	TDNN (x-vector)	Softmax	1500	TAP	VoxCeleb1	4.70	–
Okabe et al. (2018)	TDNN (x-vector)	Softmax	1500	SAP	VoxCeleb1	4.19	–
Okabe et al. (2018)	TDNN (x-vector)	Softmax	1500	ASP	VoxCeleb1	3.85	–
Hajibabaei and Dai (2018)	ResNet20	A-Softmax	128	TAP	VoxCeleb1	4.40	–
Hajibabaei and Dai (2018)	ResNet20	AM-Softmax	128	TAP	VoxCeleb1	4.30	–
Snyder et al. (2018)	TDNN (x-vector)	Softmax	1500	SP	VoxCeleb1 MUSAN RIR_NOISES	3.10	0.33
Ours (ICASSP19 Xie et al. (2019))	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	3.22	0.35
Ours + Relation Module	Thin-ResNet-34	Softmax	512	GhostVLAD	VoxCeleb2	**2.87**	**0.31**

[6] https://www.nist.gov/sites/default/files/documents/2016/10/07/sre16_eval_plan_v1.3.pdf, NIST 2016 Speaker Recognition Evaluation Plan.

7.1.6. Comparison with state-of-the-art models

In Table 10, we compare with the recent state-of-the-art models based on TDNN and x-vectors (Snyder et al., 2018) on the standard VoxCeleb1 test set. While only training on Voxceleb2, our Thin-ResNet-34 (Xie et al., 2019) achieves comparable performance to the model based on x-vectors trained on Voxceleb1&2 (EER 3.2 vs. EER 3.1).

As a fair comparison to our original model (Xie et al., 2019), we only train the relation module on the Voxceleb2 dataset. However, as explored in previous sections, we expect that incorporating both Voxceleb1 and Voxceleb2 can further boost the performance of all of our models. Overall, our Thin-ResNet with a GhostVLAD layer and a relation module currently holds the state-of-the-art result on the VoxCeleb1 dataset (Table 10).

8. Conclusion

In this paper we have introduced a scalable method to automatically generate a speaker recognition dataset, and used it to produce the VoxCeleb1 and VoxCeleb2 datasets, which are several times larger than any other speaker recognition dataset. These datasets have become a standard for the speech community to train and evaluate speaker recognition performance on. They have also played a large part in the recent NIST-SRE challenge in 2018. As mentioned by Lee et al. (2019), introducing the âVAST partitionâ in SRE18, comprising the VoxCeleb and SITW datasets, represents a 'new initiative towards speaker recognition in the wildâ, since 'a signature feature of the VAST partition is multi-speaker conversation with considerable background noise.â The VoxCeleb datasets are also the subject of the first VoxSRC challenge to be held at Interspeech 2019. We believe that the use of these datasets in challenges has allowed a paradigm shift in speaker recognition efforts in the community, encouraging the development of systems under noisy and 'in-the-wild' conditions.

We have also introduced new architectures and training strategies for the task of speaker verification. Our learnt identity embeddings are compact ($512D$) and hence easy to store and useful for other tasks such as diarisation and retrieval. The relation module, also introduced in this paper, has been shown to outperform all previous models by a significant margin on the VoxCeleb1 dataset.

Whilst our models are based on 2D convolutions applied to spectrogram inputs, further work will involve investigating alternatives that may be more efficient, such as 1D time convolutions with the frequencies of the spectrogram arranged as input channels, or 1D convolutions applied to raw waveforms directly.

We have publicly released all code, models and data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

References

Afouras, T., Chung, J.S., Zisserman, A., 2018. The conversation: deep audio-visual speech enhancement. INTERSPEECH.

Albanie, S., Nagrani, A., Vedaldi, A., Zisserman, A., 2018. Emotion recognition in speech using cross-modal transfer in the wild. In: Proceedings of the 26th ACM International Conference on Multimedia. Seoul, Republic of Korea, pp. 292–301. http://doi.acm.org/10.1145/3240508.3240578.

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J., 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Bell, P., Gales, M.J., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., et al., 2015. The MGB challenge: evaluating multi-genre broadcast media recognition. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding.

Bhattacharya, G., Alam, J., Kenny, P., 2017. Deep speaker embeddings for short-duration speaker verification. INTERSPEECH.

Cai, W., Cai, Z., Zhang, X., Wang, X., Li, M., 2018a. A novel learnable dictionary encoding layer for end-to-end language identification. arXiv preprint arXiv:1804.00385.

Cai, W., Chen, J., Li, M., 2018b. Analysis of length normalization in end-to-end speaker verification system. arXiv preprint arXiv:1806.03209.

Cai, W., Chen, J., Li, M., 2018c. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. arXiv preprint arXiv:1804.05160, 2018.

Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A., 2018. VGGFace2: a dataset for recognising faces across pose and age. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition.

Chakravarty, P., Tuytelaars, T., 2016. Cross-modal supervision for learning active speaker detection in video. In: Proceedings of the European Conference on Computer Vision.

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Return of the devil in the details: Delving deep into convolutional nets. In: Proceedings of the British Machine Vision Conference.

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

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Chowdhury, F., Wang, Q., Moreno, I. L., Wan, L., 2017. Attention-based models for text-dependent speaker verification. arXiv preprint arXiv:1710.10470.

Chung, J.S., Jamaludin, A., Zisserman, A., 2017. You said that? In: Proceedings of the British Machine Vision Conference.

Chung, J.S., Nagrani, A., Zisserman, A., 2018. Voxceleb2: Deep speaker recognition. INTERSPEECH.

Chung, J.S., Zisserman, A., 2018. Learning to lip read words by watching videos. Compu. Vis. Image Underst. 173, 76–85.

Chung, J.S., Zisserman, A., 2016a. Lip reading in the wild. In: Proceedings of the Asian Conference on Computer Vision.

Chung, J.S., Zisserman, A., 2016b. Out of time: automated lip sync in the wild. In: Proceedings of the Workshop on Multi-view Lip-reading, ACCV.

Chung, J.S., Zisserman, A., 2017. Lip reading in profile. In: Proceedings of the British Machine Vision Conference.

Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W.T., Rubinstein, M., 2018. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. SIGRAPH.

Everingham, M., Sivic, J., Zisserman, A., 2009. Taking the bite out of automatic naming of characters in TV video. Image Vis. Comput. 2009 27 (5).

Feng, L., Hansen, L.K., 2005. A new database for speaker recognition. Technical Report.

Fisher, W.M., Doddington, G.R., Goudie-Marshall, K.M., 1986. The DARPA speech recognition research database: specifications and status. In: Proceedings of the DARPA Workshop on speech recognition, pp. 93–99.

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

Ghalehjegh, S.H., Rose, R.C., 2015. Deep bottleneck features for i-vector based text-independent speaker verification. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).

Godfrey, J.J., Holliman, E.C., McDaniel, J., 1992. Switchboard: telephone speech corpus for research and development. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Greenberg, C. S., 2012. The NIST year 2012 speaker recognition evaluation plan. NIST, Technical Report.

Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J., 2016. MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Proceedings of the European Conference on Computer Vision.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Hajibabaei, M., Dai, D., 2018. Unified hypersphere embedding for speaker recognition. arXiv preprint arXiv:1807.08312, 2018.

Hansen, J.H., Hasan, T., 2015. Speaker recognition by machines and humans: a tutorial review. IEEE Signal Process. Mag. 32 (6), 74–99.

Hansen, J.H., Sarikaya, R., Yapanel, U.H., Pellom, B.L., 2001. Robust speech recognition in noise: an evaluation using the spine corpus.. INTERSPEECH.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Heigold, G., Moreno, I., Bengio, S., Shazeer, N., 2016. End-to-end text-dependent speaker verification. In: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5115–5119.

Hennebert, J., Melin, H., Petrovska, D., Genoud, D., 2000. POLYCOST: a telephone-speech database for speaker recognition. Speech Commun. 31 (2), 265–270.

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

Ioffe, S., 2006. Probabilistic linear discriminant analysis. In: Proceedings of the European Conference on Computer Vision. Springer, pp. 531–542.

Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., et al., 2003. The ICSI meeting corpus. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Karras, T., Aila, T., Laine, S., Herva, A., Lehtinen, J., 2017. Audio-driven facial animation by joint end-to-end learning of pose and emotion. ACM Trans. Graph. (TOG) 36 (4), 94.

Kazemi, V., Sullivan, J., 2014. One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874.

Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E., 2016. The megaface benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Kenny, P., 2005. Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms. CRIM, Montreal.

King, D.E., 2009. Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. 10, 1755–1758.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 1106–1114.

Lee, K. A., Hautamaki, V., Kinnunen, T., Yamamoto, H., Okabe, K., Vestman, V., Huang, J., Ding, G., Sun, H., Larcher, A., et al., 2019. I4u submission to NIST SRE 2018: Leveraging from a decade of shared experiences. arXiv preprint arXiv:1904.07386.

Lei, Y., Scheffer, N., Ferrer, L., McLaren, M., 2014. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

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

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: single shot multibox detector. In: Proceedings of the European Conference on Computer Vision. Springer, pp. 21–37.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L., 2017. Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

McCool, C., Marcel, S., 2009. Mobio database for the ICPR 2010 face and speech competition. Technical Report. IDIAP.

McCowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., et al., 2005. The AMI meeting corpus. In: Proceedings of the International Conference on Methods and Techniques in Behavioral Research, 88, .

McLaren, M., Ferrer, L., Castan, D., Lawson, A., 2016. The speakers in the wild (SITW) speaker recognition database. INTERSPEECH.

Millar, J.B., Vonwiller, J.P., Harrington, J.M., Dermody, P.J., 1994. The Australian national database of spoken language. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Morrison, G., Zhang, C., Enzinger, E., Ochoa, F., Bleach, D., Johnson, M., Folkes, B., De Souza, S., Cummins, N., Chow, D., 2015. Forensic database of voice recordings of 500+ Australian English speakers. URL:http://databases.forensic-voice-comparison.net, 2015.

Nagrani, A., Albanie, S., Zisserman, A., 2018a. Learnable pins: cross-modal embeddings for person identity. In: Proceedings of the European Conference on Computer Vision.

Nagrani, A., Albanie, S., Zisserman, A., 2018b. Seeing voices and hearing faces: cross-modal biometric matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Nagrani, A., Chung, J.S., Zisserman, A., 2017. VoxCeleb: a large-scale speaker identification dataset. INTERSPEECH.

Okabe, K., Koshinaka, T., Shinoda, K., 2018. Attentive statistics pooling for deep speaker embedding. arXiv preprint arXiv:1803.10963, 2018.

Parkhi, O.M., Vedaldi, A., Zisserman, A., 2015. Deep face recognition. In: Proceedings of the British Machine Vision Conference.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted gaussian mixture models. Dig. Signal Process. 10 (1−3), 19–41.

Reynolds, D.A., Rose, R.C., 1995. Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Trans. Speech Audio Process. 3 (1), 72–83.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, S., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Li, F., 2015. Imagenet large scale visual recognition challenge. In: Proceedings of the International Journal of Computer Vision, 2015.

Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T., 2017. A simple neural network module for relational reasoning. In: Proceedings of the Advances in Neural Information Processing Systems.

Shon, S., Tang, H., Glass, J., 2018. Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model. arXiv preprint arXiv:1809.04437.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations.

Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S., 2017. Deep neural network embeddings for text-independent speaker verification. INTERSPEECH.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: Robust dnn embeddings for speaker recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.

Song, H. O., Xiang, Y., Jegelka, S., Savarese, S., 2016. Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Stoll, L. L., 2011. Finding difficult speakers in automatic speaker recognition. Technical Report No. UCB/EECS-2011-152.

Sung, K.-K., 1996. Learning and example selection for object and pattern detection.(Ph.D. thesis)

Variani, E., Lei, X., McDermott, E., Moreno, I.L., Gonzalez-Dominguez, J., 2014. Deep neural networks for small footprint text-dependent speaker verification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Vedaldi, A., Lenc, K.,. Matconvnet − convolutional neural networks for matlab. CoRR, abs/1412.4564, 2014.

van der Vloed, D., Bouten, J., van Leeuwen, D.A., 2014. NFI-FRITS: a forensic speaker recognition database and some first experiments. In: Proceedings of the Speaker and Language Recognition Workshop.

Wan, L., Wang, Q., Papir, A., Moreno, I., 2018. Generalized end-to-end loss for speaker verification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Wang, F., Liu, W., Liu, H., Cheng, J., 2018. Additive margin softmax for face verification. arXiv preprint arXiv:1801.05599, 2018.

Woo, R., Park, A., Hazen, T.J., 2006. The MIT Mobile Device Speaker Verification Corpus: Data collection and preliminary experiments. In: Proceedings of the Speaker and Language Recognition Workshop.

Xie, W., Nagrani, A., Chung, J.S., Zisserman, A., 2019. Utterance-level aggregation for speaker recognition in the wild. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Xie, W., Shen, L., Zisserman, A., 2018. Comparator networks. In: Proceedings of the European Conference on Computer Vision.

Yapanel, U.H., Zhang, X., Hansen, J.H., 2002. High performance digit recognition in real car environments. INTERSPEECH.

Zhong, Y., Arandjelović, R., Zisserman, A., 2018. Ghostvlad for set-based face recognition. In: Proceedings of the Asian Conference on Computer Vision.