WHO DO I SOUND LIKE?
SHOWCASING SPEAKER RECOGNITION TECHNOLOGY BY YOUTUBE VOICE SEARCH

Ville Vestman, Bilal Soomro, Anssi Kanervisto, Ville Hautamäki, Tomi Kinnunen

School of Computing, University of Eastern Finland

ABSTRACT

The popularization of science can often be disregarded by scientists as it may be challenging to put highly sophisticated research into words that general public can understand. This work aims to help presenting speaker recognition research to public by proposing a publicly appealing concept for showcasing recognition systems. We leverage data from YouTube and use it in a large-scale voice search web application that finds the celebrity voices that best match to the user's voice. The concept was tested in a public event as well as "in the wild" and the received feedback was mostly positive. The i-vector based speaker identification back end was found to be fast (665 ms per request) and had a high identification accuracy (93%) for the YouTube target speakers. To help other researchers to develop the idea further, we share the source codes of the web platform used for the demo at https://github.com/bilalsoomro/speech-demo-platform.

***Index Terms*—** Large-scale speaker identification, speaker ranking, public demo, VoxCeleb, web service

1. INTRODUCTION

As methodology researchers, we often find it challenging to explain intuitively where and how our research advancements in speaker recognition can be used. To demonstrate speaker recognition technology in an appealing way to the public, many challenges need to be resolved. Besides the standard challenges of speaker recognition technology such as background noise [1], channel mismatch [2], and the requirement of fast response times in large-scale recognition tasks [3], there are challenges related to the demo design itself. First, the traditional speaker recognition setting requires at least two separate speech inputs from the user, one is for enrollment and the other one for test. The requirement of two separate recordings can be inconvenient for an user who wants to quickly test the system. The second challenge in showcasing is how to give an attractive feedback to the user. This could be implemented as a real-life application, for example, by using user's voice to open a physical lock, or in a less involved way by displaying recognition scores in a screen [4].

In this work, we present a concept for creating publicly appealing demos to showcase speaker recognition technology by leveraging public-domain target speaker data collected from YouTube. The core idea is to compare users' speech to the ones of celebrities on YouTube, who have been enrolled prior to the real-time demonstration. The results of the comparison are then displayed as a selection of YouTube videos from the best matching celebrities, which allows users to see and listen to the celebrity speakers who they most resemble to (Figure 1). Even if we focus on speaker recognition research, the same concept could also be applied for other things that can be inferred or estimated from speech such as age, emotion, or language.

This research was partially funded by the Academy of Finland (grants #313970 and #309629).

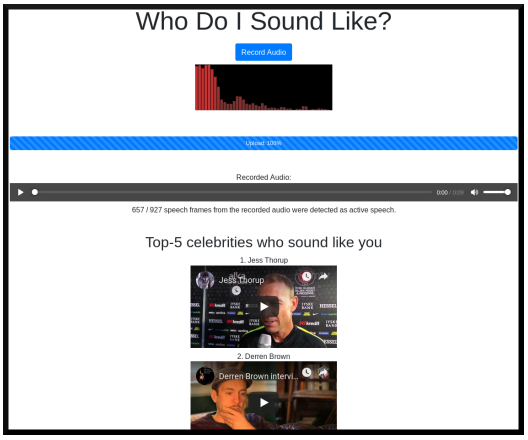

Fig. 1. A screenshot from our voice search web application displaying the basic elements of the UI: Recording button, audio visualization, playback option for the recorded speech, and the results.

For example, if the user records angry voice, the results could show YouTube videos of angry people. When the results include famous public figures, the user's interest and satisfaction of the demo tends to naturally rise. We saw this positive effect while presenting the demo in a locally organized sub-event of an European-wide "The European Researchers' Night 2018" [1] that aims to bring scientific research to public.

We run our demo on a web platform that can be used on PCs and mobile phones with an internet connection to ensure good accessibility of the demo. The web platform communicates with a computation server that runs the speaker recognition back end based on our recent work on computationally efficient i-vector extraction [5]. The back end provides the results to the web platform that displays them by using embedded YouTube video players.

Our extensive use of YouTube data has been made possible by recent automated speech data collection efforts in [6] and [7] resulting in VoxCeleb1 and VoxCeleb2 corpora, respectively. These corpora provide a large set of annotated YouTube speech data including metadata for obtaining web links to the original YouTube videos.

To best of our knowledge, prior existing speaker recognition demos have not utilized VoxCeleb data in the proposed way. We are aware of a website [2] with a similar idea, but unfortunately we have not been able to successfully run the demo to see how it functions. Based on the celebrity speaker names, that demo does not utilize VoxCeleb data, and likely does not display YouTube videos in the results.

In summary, the current work describes a novel concept that al-

[1] https://ec.europa.eu/research/mariecurieactions/actions/european-researchers-night

[2] https://celebsoundalike.com

lows speech technology research teams to demonstrate their research without requiring large amount of additional work. To help other researchers to apply the concept for their own research, we share the source code of our web platform allowing a quick start for prototyping possible demo applications. We tested the concept among the public using our voice comparison demo utilizing standard speaker recognition techniques, and the received feedback from the people was mainly positive.

2. WEB PLATFORM FOR SPEECH DEMOS

We designed the web platform with the sole purpose of demonstrating speech processing systems to the public, and in this work we used it to demonstrate speaker recognition using YouTube data. The platform is implemented as a web service in PHP and JavaScript, supporting different browser and devices. Users can select one of speech processing "methods" defined by the host of the platform. The methods could, for example, perform speaker recognition or age estimation from the recorded speech.

The back end of the platform is implemented in PHP, and thus only needs a web server capable of running PHP (*e.g.* Apache or Nginx). For simplicity and reliability, server-side code only receives WAV audio files from the clients and runs a specified method as a `system()` call, and finally returns the results to the client. For privacy reasons, the audio file is not stored on the server, and is immediately removed at the end of handling user's request. The platform supports including additional user inputs required for the analysis, *e.g.* the claimed identity for speaker verification demo.

The front end of the platform is implemented in JavaScript, which also handles recording of the audio in raw format at 16kHz. Features required by this code are supported by the most PCs and Android phones, making it easier to share the demo with others. The user interface records a sample of user's speech, queries what speech processing method should be applied on the recorded sample, sends the sample to server to be processed, and displays the results.

We share the source code of the platform in the hopes it will support other researchers in speech analysis to demonstrate their work to the public. The code includes instructions how to setup the server in couple of steps. New speech analysis systems for demonstration can be added by modifying a single JSON file.

3. SPEAKER RECOGNITION BACK END

The system comparing user's voice to voices in YouTube videos can be regarded as a *closed set speaker identification* system. As we only utilize a closed set of YouTube target speakers, we can include the data from the target speakers in the system development. In this section, we describe the data sets and the speaker identification system used for providing functionality to the web front end.

3.1. YouTube data: VoxCeleb1 & VoxCeleb2

The audio-visual VoxCeleb corpora [6, 7] have been adopted in many application areas including speaker recognition [6, 7], speech separation [8], and emotion recognition [9] to name a few. The VoxCeleb data has been automatically collected from YouTube by exploiting face verification and active speaker detection systems. An automated pipeline enabled collecting very large scale speaker recognition data sets: When combined, the VoxCeleb corpora consist of almost 1.3 million speech clips from over 170,000 YouTube videos from more than 7000 speakers and, in total, nearly 3000 hours of speech material. The average length of speech clips in VoxCeleb is about eight seconds.

The metadata provided with the VoxCeleb corpora includes, for example, speakers' names, IDs of the original YouTube videos, and the starting and ending times of the clips within the videos expressed as frames. This metadata is enough for setting up a demo where users can find best matching voices to theirs from YouTube. Although the metadata is automatically obtained, it is, in our experience, fairly accurate. Regarding to the correctness of the labels, the authors of VoxCeleb mention that the VoxCeleb2 corpus is mainly intended to be used as a training data set and that during the data collection thresholds for discarding false positives were not as strictly set as with VoxCeleb1 data collection [7]. We have witnessed a few labeling errors in VoxCeleb2, such as Finnish president Tarja Halonen being confused to talk-show host Conan O'Brien. However, the errors do not exist to an extent that would be a considerable problem for our application.

3.2. Speaker identification system description

The acoustic feature vectors of the speaker identification system consist of 20 MFCCs plus their delta and double-delta coefficients. The system discards non-speech frames using a energy based speech activity detector and normalizes obtained features to have zero mean and unit variance.

For training the system components and enrolling the celebrities, we used those speakers from VoxCeleb corpora who had more than five utterances of length of five seconds or more. There are 903,498 such utterances and 7,363 such speakers. In the training of some system components, only a fraction of this data was needed to reach close to optimal recognition accuracy. We trained an universal background model (UBM) using one-thirtieth of the selected 903,498 utterances. The UBM is a 1024-component Gaussian mixture model (GMM) [10], which is used to compute sufficient statistics for i-vector extraction. We compute 800-dimensional i-vectors by compressing mean supervectors of maximum a posteriori (MAP) adapted GMMs using probabilistic principal component analysis (PPCA) as described in [5]. This is a (speed-wise) high-performing alternative to the stardard i-vector extraction that is traditionally done via front-end factor analysis [11, 12]. We trained the PPCA model using one-fifteenth of the selected data.

Prior to scoring, i-vectors are centered using the mean computed from the whole training data of 903,498 utterances and then normalized to unit length. Scoring is performed with a simplified Gaussian probabilistic linear discriminant analysis (G-PLDA) model [13], which has a 350-dimensional speaker subspace. The G-PLDA model was trained using the whole training data.

At the online stage, the i-vector extracted from user's recording is scored against all of the 903,498 i-vectors used in PLDA training. The speakers are sorted according to the scores of their highest scoring utterances, from highest score to lowest. Finally, the system sends the names of the top-5 speakers together with the links to the YouTube-videos that correspond to the highest scoring utterances to the client.

3.3. System runtime considerations at online stage

To ensure fast response times, we implemented the speaker recognition back end as a server that has all the necessary models preloaded in the memory. The server is implemented with Python using scientific computing libraries available to it (*e.g.* NumPy and SciPy). We pay special attention to the PLDA scoring and i-vector extraction as they are the most time consuming steps during the computation.

In [13], it is shown that the score for a trial using G-PLDA can be computed as

$$\text{score} = \tilde{\boldsymbol{\eta}}_1^\intercal \widetilde{Q} \tilde{\boldsymbol{\eta}}_1 + \tilde{\boldsymbol{\eta}}_2^\intercal \widetilde{Q} \tilde{\boldsymbol{\eta}}_2 + 2\tilde{\boldsymbol{\eta}}_1^\intercal \Lambda \tilde{\boldsymbol{\eta}}_2 + \text{const},$$

Table 1. Speaker rank testing for six public figures using 10 audio clips from each speaker. The speaker ranks range from 1 to 5 and 'x' is shown if the result list of top-5 speakers did not contain the correct speaker at all. The tests are performed with and without a replay channel. The replay experiment does not require direct access to the back end system, but can be done by using the web demo only.

	Without replay channel				With replay channel			
		Occurrences in				Occurrences in		
Speaker's name	list positions for 10 clips	top1	top3	top5	list positions for 10 clips	top1	top3	top5
Hillary Clinton	1111111111	10	10	10	1111111111	10	10	10
Ariana Grande	1111111111	10	10	10	3111111111	9	10	10
Oprah Winfrey	1111111111	10	10	10	1311111112	8	10	10
Johnny Depp	1111112112	8	10	10	1111x11121	8	9	9
Bruno Mars	1411111112	8	9	10	1x21111211	7	9	9
Conan O'Brien	1111111111	10	10	10	1111111111	10	10	10
Total (in % of max.)		93	98	100		87	97	97

where $\tilde{\boldsymbol{\eta}}_1$ and $\tilde{\boldsymbol{\eta}}_2$ are lower dimensional projections of enrollment and test i-vectors, respectively, and where $\tilde{\boldsymbol{\eta}}_1^\intercal \widetilde{Q} \tilde{\boldsymbol{\eta}}_1$ and $\tilde{\boldsymbol{\eta}}_1^\intercal \Lambda$ can be precomputed.

As we work with an identification system (one test segment vs. all enrollment segments), the second term $\tilde{\boldsymbol{\eta}}_2^\intercal \widetilde{Q} \tilde{\boldsymbol{\eta}}_2$ is a constant and thus can be neglected. Therefore, to get all the $n = 903{,}498$ scores at online stage, we only need to compute

$$\text{scores} = \boldsymbol{\nu} + 2DP\boldsymbol{\eta}_2,$$

where $\boldsymbol{\nu}$ is an n-dimensional vector containing precomputed values $\tilde{\boldsymbol{\eta}}_1^\intercal \widetilde{Q} \tilde{\boldsymbol{\eta}}_1$, matrix $D \in \mathbb{R}^{n \times 350}$ contains precomputed vectors $\tilde{\boldsymbol{\eta}}_1^\intercal \Lambda$, and P is a 350×800 projection matrix that projects test i-vector $\boldsymbol{\eta}_2$ to a lower dimensional space so that $\tilde{\boldsymbol{\eta}}_2 = P\boldsymbol{\eta}_2$. The product $D\tilde{\boldsymbol{\eta}}_2$ can be efficiently parallelized.

The i-vector extraction using PPCA is simply a matter of compressing 61440-dimensional GMM-supervector to 800-dimensional space using a precomputed projection matrix. Note that the traditional approach for i-vector extraction would, in addition, require inverting an 800×800 posterior covariance matrix [14, 5].

4. SYSTEM EVALUATION

We tested our voice search demo and the underlying speaker recognition back end in multiple ways using both objective and subjective measures in evaluation. On the objective side, we computed an equal error rate (EER) using VoxCeleb speaker verification protocol and further we tested the rankings that the system displays for newly downloaded and replayed YouTube data. On the subjective side, we gathered feedback from the users of the system, including their opinions on how close the displayed top five celebrities sound to the user.

4.1. Evaluation using VoxCeleb speaker verification protocol

The VoxCeleb1 speaker verification test protocol includes 37720 trials with a balanced number of same speaker trials and impostor trials. The trial list has been formed using 4715 utterances from 40 speakers. Using this protocol, we obtained EER of 6.69 %. This result is better than the baseline result for i-vectors in [7], but should not be directly compared as our system utilizes testing utterances also in system training.

4.2. Speaker rank testing on non-VoxCeleb YouTube data

To test the the final deployed demo, we studied the speaker rankings the system outputs. For this purpose, we collected a small set of new YouTube data. This set contains 10 new speech clips for six public figures in VoxCeleb corpora. The clips are about 15 seconds long each and they are extracted from videos that are not already present in VoxCeleb corpora. When the new clips are fed to the speaker recognition back end, the output lists of top-5 speakers should contain the correct speaker as they are present in VoxCeleb and hence are already enrolled to the system.

The new test data was used with the system in two ways: First, we downloaded the speech clips from YouTube and fed the data directly to the speaker recognition back end. Secondly, we played files directly from YouTube and at the same time recorded them with the web demo. Unlike the first approach, the second one includes the channel effects caused by replaying the data. In the replay experiment, the playback device was Sony SRS-XB10 portable Bluetooth speaker while the web demo was ran in Chrome browser in Nokia 8 smartphone running Android 8.1.0. The distance between the two devices was kept to 5 cm as the recording device was held by hand above the up facing speaker. The room in which the experiment took place was quiet and the only background noise that was present was the fan noise of the laptop which was connected to the speaker.

For both settings, with and without replay, the speaker rankings for all the test utterances are shown in Table 1. In addition, the table contains statistics of the number of occurrences in the top-1, top-3, and top-5 rankings. Without the replay, the system was always able to include the correct speaker to the top-5 list and 93% of the times the speaker was identified correctly (*i.e.*, in top-1). Replaying the audio clips decreased the system performance only slightly as the correct speaker was left outside the top-5 list only twice out of the 60 trials.

To get insight of how long of an utterance is required for getting good results in our celebrity matching demo, we studied the effect of length of the test utterance on system accuracy. We ran the previous experiment without the replay effect using utterances clipped to lengths ranging from 1 second to 15 seconds. We found that the test segment needs to be at least 9 seconds to obtain close to optimal performance and at least 5 seconds to obtain identification accuracies greater than 70% (Figure 2).

4.3. Feedback and impressions from public testing

The first public test for our voice search demo took place in the event "The European Researchers' Night 2018" (September 28, 2018), where researcher's from many fields were displaying their research to the public. The event was funded by EU and it was organized in many countries across the Europe. In our local event, we were

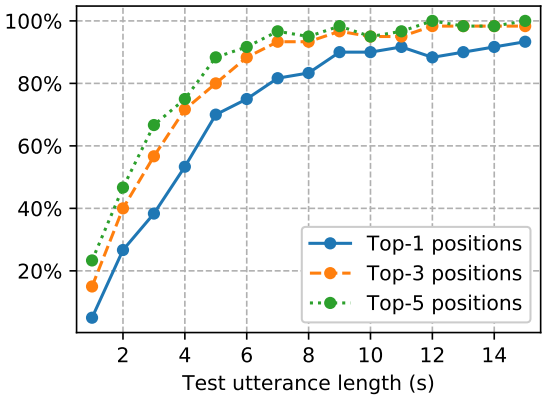

Fig. 2. The effect of utterance length on speaker ranking performance. Specifically, the graph shows how often the target speakers are displayed in the top-lists when tested with different lengths of test utterances from the target. An utterance of length 9s is required to reach close to optimal performance.

Table 2. Computation times for the different steps in the voice comparison pipeline. The steps marked with * are parallelized to 16 CPU cores while others steps utilize only 1 core. The total response time is the time it takes to upload the speech and compute and display the results. The data was collected from 402 requests, except for the total response time which was collected together with the feedback questionnaire (n=27).

	Times in milliseconds (ms)		
	median	mean	SD
Audio loading, MFCC extraction	47.1	86.2	142.1
Sufficient statistics computation	20.9	46.3	98.1
MAP adaptation	0.8	0.9	0.6
Supervector compression (PPCA)*	42.6	56.5	28.3
I-vector centering & length norm.	0.1	0.1	< 0.1
I-vector compression (PLDA)	0.4	0.5	0.3
PLDA scoring*	336.4	423.8	195.8
Sorting speakers	39.6	44.6	13.8
Total time in computing server	521.5	661.0	331.6
Total response time	1791.1	2503.5	1975.1

showcasing our demo for five hours and for the most of the time there was a long queue of people waiting for their turn to test our demo. In total, approximately 150 people tried the demo. The feedback was mostly positive, although not everyone was satisfied with their results. As the event was targeted for families, many of the testers were children. This was a slight problem as only a small minority of the speakers in VoxCeleb corpora are children, causing it to be difficult to find a good voice match for everyone.

In the event, we were using our own high-quality microphone (Zoom H6 Handy Recorder, XY mic) and a laptop that was well tested with the demo. To see how the demo works "in the wild", we shared a web link to our demo in a multiple social media platforms. The shared demo application was equipped with a short feedback questionnaire for subjective evaluation. We also collected error reports containing system information of the devices on which the demo did not work.

The public testing revealed that the device and browser support is still quite limited due to some issues with the audio recording and

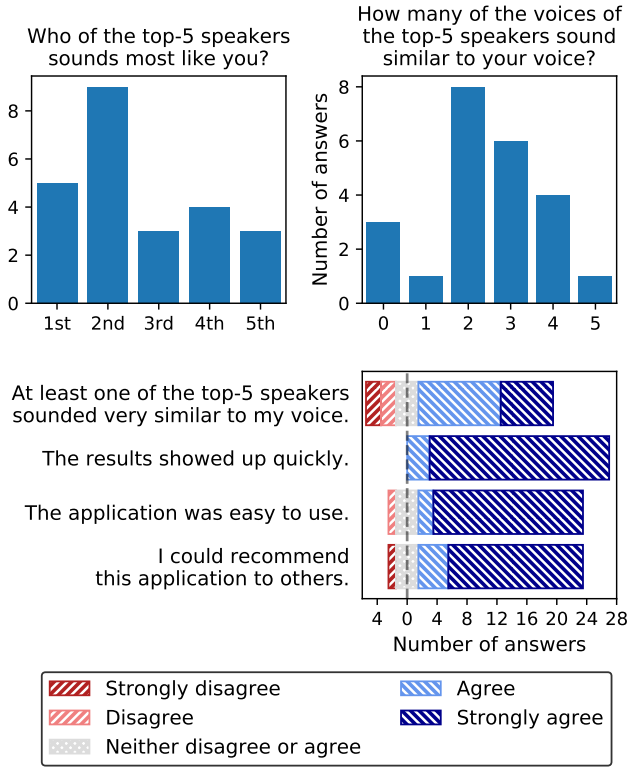

Fig. 3. Results from the feedback questionnaire, gathered from users using platform that finds matches for their speech from a set of over 7000 celebrities. Based on these subjective assessments, the system is able to find good matches for users' speech in most cases.

playback support. Based on the feedback, we estimate that demo ran on 50 to 70 percent of the device-browser configurations that our test users were using. We also got some good suggestions how to improve the user interface and we believe that together with improved browser support the user experience can be very good as the received answers (n=27) to the questionnaire were already fairly positive as can be seen from Figure 3.

4.4. Response and computation times

During the test in the wild, we collected computation times of the different steps in the voice comparison. The statistics are summarized in Table 2. The average time to compute one voice comparison request was 661 milliseconds, which means that our computation server could, theoretically, respond to 5000 requests in an hour without processing multiple requests in parallel. The total response time, on average, was about 2.5 seconds. As seen from Figure 3, this level of responding speed was considered to be fast.

5. CONCLUSIONS

We successfully capitalized the appeal to public figures with our YouTube voice search demo application. The objective and the subjective evaluations of the demo showed that the platform was mostly successful in providing good results and also being convenient to use. The feedback received from the users allows us to further develop our demo platform, which we have shared for open source development at https://github.com/bilalsoomro/speech-demo-platform. We would be happy to see the proposed concept to be applied in the future with other speech related recognition systems as well.

6. REFERENCES

[1] Xiaojia Zhao, Yuxuan Wang, and DeLiang Wang, "Robust speaker identification in noisy and reverberant conditions," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 836–845, 2014.

[2] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, March 2005, vol. 1, pp. I/629–I/632 Vol. 1.

[3] L. Schmidt, M. Sharifi, and I. L. Moreno, "Large-scale speaker identification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1650–1654.

[4] Kong Aik Lee, Anthony Larcher, Helen Thai, Bin Ma, and Haizhou Li, "Joint application of speech and speaker recognition for automation and security in smart home," in *Proc. Interspeech*, 2011.

[5] Ville Vestman and Tomi Kinnunen, "Supervector compression strategies to speed up i-vector system development," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 357–364.

[6] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[8] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman, "The conversation: Deep audio-visual speech enhancement," in *Proc. Interspeech*, 2018, pp. 3244–3248.

[9] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman, "Emotion recognition in speech using cross-modal transfer in the wild," in *ACM Multimedia*, 2018.

[10] Douglas Reynolds, Thomas Quatieri, and Robert Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[11] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre 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] Patrick Kenny, "A small footprint i-vector extractor," in *Odyssey*, 2012, vol. 2012, pp. 1–6.

[13] Daniel Garcia-Romero and Carol Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[14] Srikanth Madikeri, "A fast and scalable hybrid FA/PPCA-based framework for speaker recognition," *Digital Signal Processing*, vol. 32, pp. 137–145, 2014.