Robust Deep Feature for Spoofing Detection - The SJTU System for ASVspoof 2015 Challenge

Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

bobchennan@gmail.com, yanminqian@sjtu.edu.cn, richman@sjtu.edu.cn
bobmilk@sjtu.edu.cn, kai.yu@sjtu.edu.cn

Abstract

Recently there have been wide interests in speaker verification for various applications. Although the reported equal error rate (EER) is relatively low, many evidences show that the present speaker verification technologies can be susceptible to malicious spoofing attacks. Inspired by the great success of deep learning in the automatic speech recognition, deep neural network (DNN) based approaches are developed on the spoofing detection for the first time. In this paper, a novel DNN based robust representation is proposed for the spoofing detection to extract the representative spoofing-vector (s-vector). Then the mahalanobis distance and appropriate normalization methods are investigated to get the best system performance. Using the designed deep learning based strategy, our team obtained an impressive result on spoofing detection task, and achieved the 3rd position in the first spoofing detection challenge evaluation, i.e. ASVspoof 2015 Challenge.

Index Terms: Automatic speaker verification, Spoofing attack, Anti-Spoofing, Spoofing detection, Deep learning

1. Introduction

Biometric recognition is a broad field going from the classic fingerprint to face recognition and nowadays speech is naturally used to restrict access to certain areas. Speaker verification is therefore one of the crucial ways of guarding access to data. The main focus of speaker verification is to detect whether the (real) speaker, who registered himself with the system, did produce an utterance to require access to a system or if that utterance was produced by an impostor. The speaker verification has got a lot of research attentions in recent years and have been shown to offer promising performance in smartphone logical access scenarios [1] and e-commerce.

Although it is widely acknowledged that biometric systems can be ”spoofed”[2, 3, 4, 5], research about securing a system against possible malicious impostors was significantly smaller. Attacks can happen mainly on two different categories. One category is the direct attacks, also refered as *spoofing attacks*, which do not need the access permission in speaker verification system. The other category is the indirect attacks, which can be applied within the speaker verification system. Since spoofing attacks are more easily to implement, it is the greatest threat to the system. There are mainly four kinds of spoofing attacks discussed in previous works:

- Impersionation
- Replay
- Speech synthesis
- Voice conversion

Impersonation generally requires experts to mimic a target speaker's voice and hence there are limited data and research in the past. Results in Wu's work [6] suggested there are no consistent result and more research are needed. Replay uses the recorded speech to spoof the system, and better features [7, 8], channel noise detection [9] are suggested to be effective. The last two kinds of attacks become increasingly easily due to the availability of many online open-source libraries. In recent research, these two attack types got a lot of attentions [10, 11, 12].

The ASVspoof challenge has been designed to simulate these cases in reality and for the first time support independent assessments of vulnerabilities to spoofing and of countermeasure performance. The ASVspoof challenge 2015 focused on the spoofing detection, which mainly includes synthesized speech or converted speech attacks detection. The challenge provides training and development data, which consists of both spoofed and natural speech, where five different spoofing algorithms were used. As an particular hard part, the (spoofed) evaluation data was only partly generated using the known techniques the same as the training and development data, therefore participant needs to design a system capable of handling known and unknown attacks (5 additional algorithms in the evaluation data).

Besides standalone spoofing detection, there are also some works combining the detection process and verification process. Elie Khoury's work [13] used the integrated PLDA system to combine these two process and the firstly applied directly to i-vector [14]. The result reveals the advantages of the combination which leads to a large improvement.

The remainder of this paper is organized as follows. Section 2 firstly reviews the previous work on the spoofing detection task, especially describes several types features reported in the spoofing detection, then gives the model which was widely used by previous works and the baseline of these features. The novel DNN based spoofing detection approach is presented in detail in Section 3, and then experimental results and analysis are described in Section 4. Finally Section 5 concludes the whole work.

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and JiangSu NSF project No.201302060012.

2. Previous works on spoofing detection

2.1. Features

As previous works [15, 16, 17, 18] suggested, feature extraction is important to detect whether spoofing was applied. For instance, low variance in Hidden Markov Model (HMM) generated speech is sensitive to higher order of mel-cepstrum analysis (MCEP) features [16]. Using this feature is sufficient to detect HMM generated speech. Furthermore the research towards better features shows that detecting synthesized speech is correlated to commonly used features in text-to-speech synthesis (TTS). Previous studies [19] show that artefacts in the phase spectrum occur when using a synthesis filter on synthesized speech. Phase spectrum based features was suggested to discriminate better than commonly used magnitude based ones, e.g. MFCC. So common TTS features was used in our baseline to gain a better intra frame discrimination. When it comes to TTS, not just one feature type, a combination of three different ones are used. These features are called mel-cepstrum analysis (MCEP), Band-Aperiodicity (BAP) and pitch (LF0). Generally a static feature size of 31 is used, which is composed of 25 dimensional MCEP, 5 dimensional BAP and 1 dimensional LF0. Specific spoofing detection features dubbed as modified group delay cepstral coefficients (MGDCC) [20] are also implemented in our baseline systems.

2.2. Model

While i-vector approach [13] got impressive performance in the combination system, it is not adopted in our experiments due to two reasons. One reason is that the task only aims at spoofing detection, and previously there are no clear evidences to show the superior of i-vector approach in standalone spoofing detection. Second, the training set is quite small and may not be enough for the i-vector training. Also the corpus consists of short utterances (2-3s) which decrease the i-vector approach performance.

As previous work [21] suggests, Gaussian Mixture Models (GMM) are adopted to detect incoming attacks. For training the given dataset was split into two parts, namely the natural and the spoofed speech. Two GMM models, named $\boldsymbol{M}_n$ for natural speech, $\boldsymbol{M}_s$ for spoofing speech are estimated. The score was then calculated as follows.

$$\text{score}(\mathbf{x}) = \log\left(P\left(\boldsymbol{M}_n|\mathbf{x}\right)\right) - \log\left(P(\boldsymbol{M}_s|\mathbf{x})\right) \qquad (1)$$

where $P\left(\boldsymbol{x}\right)$ is gaussian distributed. So natural speech tend to have higher score value, which could be compared with a threshold which was pre-estimated.

2.3. Baseline

Before training some complex models, different feature types are evaluated.A two class GMM-UBM was used as our evaluation model. A GMM with 512 mixtures was trained for each part. Table 1 gives the results on the development set using features [1] described in Section 2.1 and GMM model described in Section 2.2. The results are illustrated by equal error rate (EER), minimal detection cost function (Min. DCF), and actual detection cost function (Act.DCF). The minimum DCF is the DCF value corresponding to the threshold that minimizes it on the test data, while the actual DCF is calculated based on the evaluation the goodness of log-likelihood-ratios introduced in [22].

Table 1: Development set results with the GMM approch

Feature	EER	Act.DCF	Min. DCF
MGDCC	22.4%	0.750	0.637
BAP+MCEP	11.2%	0.551	0.356
MCEP	9.8%	0.503	0.322

3. DNN based spoofing detection

Inspired by the success of the deep neural network in speech recognition[23] and speaker verification[24], a robust DNN based spoofing detection approach is proposed for this task. From feature extraction to classification, the whole model was a DNN at its core, which maps the physical features to more discriminant ones.

Although the implementation of the previously mentioned MCEP features leads to the best results when using the GMM model for spoofing task, these features was not used directly in our neural network training. Because according to our initial experiments, using these features as inputs will lead to a bad frame accuracy during DNN training, whereas the reason still needs further investigation. Consequently traditional filter bank features with Δ (FBANK_D) was adopted, which are broadly used in the speech recognition projects[23].

3.1. Robust deep feature extraction: spoofing-vector

The core of our spoofing detection system is the deep neural network which plays the role as a robust spoofing feature extractor. Inspired by approaches like JFA[25, 26] and i-vector [14], we tried to find a compact, robust and abstract feature representation which could be used directly for the distance metrics or classifiers. The deep neural networks were used to perform such transformation, rather than the factor analysis model, which was commonly used in the normal i-vector framework.

Specifically, a supervised DNN was trained on the training data. For this challenge, there are different information available which can be used for DNN training:

- Speaker indicator, e.g. the speaker identity for the specific audio
- Spoofing indicator, e.g. whether the audio has been spoofed or what kind of spoofing techniques is used

Based on these information, the supervised training labels could be designed as several types in order to detect spoofing:

- Spoofing technique classification labels, which discriminate spoofing techniques in the training set
- Spoofing labels, which discriminate whether spoof or not

The inputs of these networks are formed by stacking each current frame with its left and right context frames. The DNN model used in the spoofing task is illustrated as Figure 1.

Once the deep neural network has been trained successfully, the outputs of last hidden layer, which give the most abstract and robust representation are used as our new feature representation. Assuming that the outputs of the last hidden layer for audio s are $\boldsymbol{X}_{s,1}$, $\boldsymbol{X}_{s,2}$, ..., $\boldsymbol{X}_{s,n}$, where n indicates frame index in audio s. As shown in expression 2, the mean value of

[1] There are no delta and LF0 features using in the baseline systems due to the some mistakes in the extraction process and lack of time during the contest.

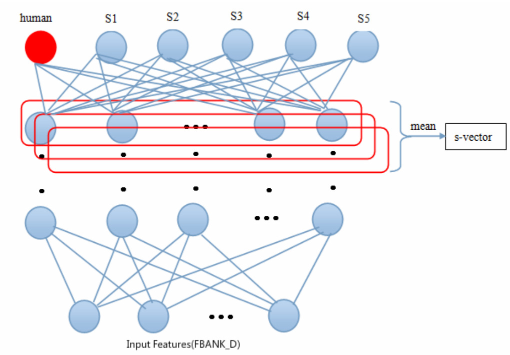

Figure 1: s-vector extraction process

these outputs was used as the final representation of audio s, which is further called *spoofing vector* (s-vector). This new robust representation is similar as the work in Google's speaker verification system [24].

$$\text{s-vector}(s) = \frac{\boldsymbol{X}_{s,1} + \boldsymbol{X}_{s,2} + ... + \boldsymbol{X}_{s,n}}{n} \tag{2}$$

3.2. Score: mahalanobis distance

If all spoofing methods in the evaluation set would appeared in the training set, then our work could be focused on the selection of proper classifiers. However, since the spoofing algorithms in both sets are not exactly the same, we focused on building a robust system. We assumed that all s-vectors within a class are normal distributed. In this example, one class indicates human voice and other classes indicates different spoofing techniques. Speech synthesis technique and voice conversion technique are all used in the training set, so spoofing discrimination on this set assumed to have powerful ability to identify these two types of attack.

Mahalanobis distance is used to estimate the distance between test segments and different classes:

$$l_c(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) \tag{3}$$

Here Σ is the average of the different classes covariance matrices estimated in training data.

Also after expansion, the first term $x^T \Sigma x$ can be omitted and the final score for each class c can be written as

$$\text{score}_c(\mathbf{x}) = \mathbf{x}^\top(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c) + \left(-\frac{1}{2}\boldsymbol{\mu}_c{}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right) \tag{4}$$

PLDA [27, 28] could be also utilized as a score method. Traditionally PLDA works well for the unseen cases, which may fit for the detection of unseen spoofing algorithms.Parameters in PLDA model are adjusted to obtain better results.

3.3. Normalization

For classification, the probability can be used as a score function, which are estimated by Bayes Theorem.

$$P(\text{class} = c|\boldsymbol{X} = \mathbf{x}) = \frac{\pi_c P(\mathbf{x}|c)}{\sum_{k=1}^{K} \pi_k P(\mathbf{x}|k)} \tag{5}$$

However, there are some "unknown classes", i.e. the unknown spoofing algorithms, which only appear in the evaluation set. Indeed different normalization methods was used to eliminate the influence from sessions and speech content.

Traditionally, test normalization (TNorm) and zero normalization (ZNorm) [29] improve the result for a speaker verification system. Here TNorm was used in our system due to the reason that different spoofing algorithms may have a score close to spoofing algorithms given in the training set. TNorm can be written as:

$$\text{score}_{\text{TNorm}}(\mathbf{x}) = \frac{(\text{score}_{\text{human}}(\mathbf{x}) - \text{mean}(\mathbf{x}))}{\text{std}(\mathbf{x})} \tag{6}$$

where mean and std function was the mean and standard variance value within $\text{score}_{S1}(\mathbf{x})$, $\text{score}_{S2}(\mathbf{x})$, $\text{score}_{S3}(\mathbf{x})$, $\text{score}_{S4}(\mathbf{x})$, $\text{score}_{S5}(\mathbf{x})$, these scores are given by five classes representing five spoofing algorithms in training set[2].

Besides after the contest we also tried another normalization strategy, named probabilistic normaliztion (PNorm), inspired by equation (5). The intuition is that other spoofing algorithms has similarities with given spoofing algorithms and can be considered as the combination of given spoofing algorithms.

$$\begin{aligned} \text{score}_{\text{PNorm}}(\mathbf{x}) &= log(\frac{\exp{(l_{\text{human}}(\mathbf{x}))}}{\sum_k \exp{(l_k(\mathbf{x}))}}) \\ &\approx \text{score}_{\text{human}}(\mathbf{x}) - \max_{k \neq \text{human}}(\text{score}_k(\mathbf{x})) \end{aligned} \tag{7}$$

where k belongs to the indexes of classes. Here each class has been assumed to have the same prior probability.

4. Experiments

In this section we are providing descriptions about the spoofing challenge and present our system and experimental results in detail.

4.1. ASVspoof challenge 2015

In the past, spoofing attacks have generally been developed with full knowledge of a particular ASV system. Similarly, countermeasures have been developed with full knowledge of the spoofing attack which they are designed to detect. The ASVspoof challenge has been designed to address old shortcomings and to support, for the first time, independent assessment of vulnerabilities to spoofing and of countermeasure performance. The first evaluation, ASVspoof 2015, is focused on the spoofing detection.

The evaluation data contains both genuine and spoofed speech. Genuine speech is collected from 106 speakers (45 male, 61 female) and with no significant channel or background noise effects. Spoofed speech is generated from the genuine data using a number of different spoofing algorithms. The full dataset is partitioned into three subsets, including training, development and evaluation dataset. There are no speaker overlap across the three subsets regarding target speakers used in voice conversion or TTS adaptation. The specific numbers are illustrated in Table 2.

The recording conditions for the evaluation data are exactly the same as those for development dataset. Spoofed data is generated according to diverse spoofing algorithms, including 5 algorithms (3 voice conversion implementations and 2 speech synthesis implementations) used to generate the development

[2] S1, S2, S5 are 3 different voice conversion systems and S3, S4 are speech synthesis systems. More specific introduction can be found in dataset explanation.

Table 2: Number of non-overlapping target speakers and utterances in the training, development and evaluation datasets

Subset	#Speakers		#Utterances	
	Male	Female	Genuine	Spoofed
Training	10	15	3750	12625
Development	15	20	3497	49875
Evaluation	20	26	9404	184000

dataset in addition to others, designated as ”unknown” spoofing algorithms.

Some traditional baseline methods have been illustrated in Table 1 described in Section 2.3.

4.2. Effect of the proposed DNN based spoofing detection

The DNN-based VAD is applied for pre-processing. It got 96.60% VAD frame accuracy on our labelled cross validation data. Then silence frames before the first speech frame and after the last speech frame were removed.

After that the features were normalized: removing the average (zero-mean) and scaling the variance (unit-variance). The RBM pre-training is utilized in this work. A 5 layers RBM network with 1024 nodes per layer was trained with 0.02 learning rate and 0.5 momentum. The weight cost was set to 0.0002. In our training recipe the first layer is trained with 20 iterations and the other layers are trained 10 iterations.

After the RBM pre-training, different networks were trained based on the same configuration. The training set was randomly partitioned into two subsets, one for network training and one for cross validation. The ratio between training and cross validation is 7:1. We used a learning rate of 2 and 0.85 momentum. The weight cost was set to 1×10^{-6} and the halving factor was set to 0.85.

The neural network aimed at discriminating spoofing techniques. The intuition is to learn a prediction function of the spoofing techniques given local context-extended features. The back-propagation was used to fine-tune these networks until the relative improvement between two iterations is negligible.

After the network training, the last layer (output layer) was dropped and the remaining layers can be used as the new feature extractor. L2-normalization was then used on the output for each frame. Each output for every frame was then averaged out into a 1024-dimensional s-vector.

Two types of classification are tried in the DNN construction for spoofing detection:

- 6 classes: human, S1, S2, S3, S4, S5 (5 algorithms given in training set)
- 2 classes: human, spoofing

Table 3 gives our result on the development set using DNN approach describe in Section 3. The result are illustrated by equal error rate (EER), minimal detection cost function (Min. DCF), and actual detection cost function (Act.DCF). The ROCCH EER [22] is used for EER calculation in this paper.

Compared with Table 1, it can be seen that although GMM approach with different features got very low EER, our new DNN-based method leads to much more better performance.

4.3. Evaluation result and analysis

For the evaluation, Our team submitted the 6 classes target based DNN with TNorm as the primary submission and Table 4 shows the final results on the evaluation data. Although TNorm normalization gets the worse result in the development set, small scaled experiments convinced that it has better performance on ”unknown attacks”. Here “known EER” indicates EER for attacks which use the same algorithms as the development set, ”unknown EER” indicates EER for attacks which were not previously seen in the development set, and ”all EER” is the EER for all data.

Table 3: Development set result with proposed DNNs

# Target	Norm	EER	Act. DCF	Min. DCF
2	—	0.013%	0.006	3.9×10^{-4}
6	`TNorm`	0.033%	0.059	9.1×10^{-4}
6	`PNorm`	0.020%	0.010	5.8×10^{-4}
6	`PLDA`	5.2%	2.386	1.7×10^{-1}

Table 4: Final result (EER(%)) on evaluation data

# Target	Norm	Known	Unknown	All
6	`TNorm`	0.058	4.998	2.528
6	`PNorm`	0.046	4.516	2.281
6	`PLDA`	8.650	20.54	14.59

The following Figures 2 show the results for the first 12 teams in the rank list. Our scores are labelled with white color. It can be seen that s-vector method has generality and robustness properties and works quite well for the unknown attacks. Our proposed deep feature, s-vector, based system get the 3^{rd} position among all 16 teams.

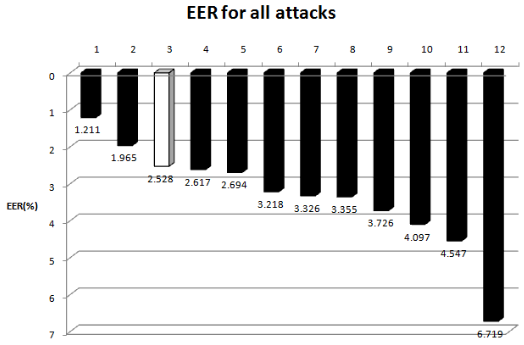

Figure 2: EER for all attacks

5. Conclusions

In this paper a new simple model which can effectively detect spoofing attacks on a speaker verification system was proposed. A deep learning framework is described for extracting useful knowledge from the audio to form compact, abstract and robust deep representation. First, a spoofing-discriminant network is employed to learn spoofing algorithms. With the learned network, utterance level average of the outputs from the last hidden layer, refereed as s-vector, is calculated. Finally mahalanobis distance with normalization is applied to s-vector. Experiments show that our proposed s-vector get good grade on the development set. The proposed system achieved the 3^{rd} position in the first spoofing detection challenge evaluation, i.e. ASVspoof 2015 Challenge.

6. References

[1] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *IEEE Signal Processing Society Speech and language Technical Committee Newsletter*, 2013.

[2] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[3] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *Proc. Interspeech*, 2013, pp. 925–929.

[4] N. Evans, T. Kinnunen, J. Yamagishi, Z. Wu, F. Alegre, and P. De Leon, "Speaker recognition anti-spoofing," in *Handbook of Biometric Anti-Spoofing*. Springer, 2014, pp. 125–146.

[5] Z. Wu and H. Li, "Voice conversion and spoofing attack on speaker verification systems," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, 2013, pp. 1–9.

[6] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: a survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[7] J. Villalba and E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems," in *Biometrics and ID Management*. Springer, 2011, pp. 274–285.

[8] ——, "Preventing replay attacks on speaker verification systems," in *Security Technology (ICCST), 2011 IEEE International Carnahan Conference on Security Technology*. IEEE, 2011, pp. 1–8.

[9] Z.-F. Wang, G. Wei, and Q.-H. He, "Channel pattern noise based playback attack detection algorithm for speaker recognition," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, vol. 4. IEEE, 2011, pp. 1708–1713.

[10] F. Alegre, R. Vipperla, A. Amehraye, and N. Evans, "A new speaker verification spoofing countermeasure based on local binary patterns," in *Proc. Interspeech*, 2013, p. 5p.

[11] Z. Kons and H. Aronowitz, "Voice transformation-based spoofing of text-dependent speaker verification systems." in *Proc. Interspeech*, 2013, pp. 945–949.

[12] Z. Wu, A. Larcher, K.-A. Lee, E. Chng, T. Kinnunen, and H. Li, "Vulnerability evaluation of speaker verification under voice conversion spoofing: the effect of text constraints." in *Proc. Interspeech*, 2013, pp. 950–954.

[13] E. Khoury, T. Kinnunen, A. Sizov, Z. Wu, and S. Marcel, "Introducing i-vectors for joint anti-spoofing and speaker verification," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[15] "An adaptive algorithm for mel-cepstral analysis of speech," *Proc. ICASSP*, vol. 1, pp. 137–140, 1992.

[16] L.-W. Chen, W. Guo, and L.-R. Dai, "Speaker Verification against Synthetic Speech," *Proceedings of the 7th International Symposium on Chinese Spoken Language Processing, ISCSLP 2010*, pp. 309–312, 2010.

[17] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of HMM-based synthetic speech," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[18] P. L. De Leon, B. Stewart, and J. Yamagishi, "Synthetic Speech Discrimination using Pitch Pattern Statistics Derived from Image Analysis." *Proc. Interspeech*, no. 1, 2012.

[19] Z.-z. Wu, E. S. Chng, and H. Li, "Detecting Converted Speech and Natural Speech for anti-Spoofing Attack in Speaker Recognition," *Proc. Interspeech*, pp. 1700–1703, 2012.

[20] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," *Proc. ICASSP*, pp. 7234–7238, 2013.

[21] Z. Wu, T. Kinnunen, and E. Chng, "A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2012.

[22] N. Brummer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2010.

[23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and K. Brian, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[24] E. Variani, X. Lei, E. Mcdermott, I. L. Moreno, and J. Gonzalez-dominguez, "DEEP NEURAL NETWORKS FOR SMALL FOOTPRINT TEXT-DEPENDENT SPEAKER VERIFICATION Google Inc ., USA ATVS-Biometric Recognition Group , Universidad Autonoma de Madrid , Spain," *Proc. ICASSP*, pp. 4080–4084, 2014.

[25] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, May 2005.

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

[27] P. Matejka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *Proc. ICASSP*. IEEE, 2011, pp. 4828–4831.

[28] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," in *Proc. ICASSP*. IEEE, 2013, pp. 7649–7653.

[29] C. Barras and J.-L. Gauvain, "Feature and score normalization for speaker verification of cellular data," in *Proc. ICASSP*, vol. 2. IEEE, 2003, pp. II–49.