Application of Convolutional Neural Networks to Speaker Recognition in Noisy Conditions

Mitchell McLaren, Yun Lei, Nicolas Scheffer, Luciana Ferrer

Speech Technology and Research Laboratory, SRI International, California, USA

{mitch,yunlei,scheffer,lferrer}@speech.sri.com

Abstract

This paper applies a convolutional neural network (CNN) trained for automatic speech recognition (ASR) to the task of speaker identification (SID). In the CNN/i-vector front end, the sufficient statistics are collected based on the outputs of the CNN as opposed to the traditional universal background model (UBM). Evaluated on heavily degraded speech data, the CNN/i-vector front end provides performance comparable to the UBM/i-vector baseline. The combination of these approaches, however, is shown to provide improvements of 26% in miss rate to considerably outperform the fusion of two different features in the traditional UBM/i-vectors approach. An analysis of the language- and channel-dependency of the CNN/i-vector approach is also provided to highlight future research directions.

Index Terms: Deep neural networks, Convolutional neural networks, Speaker recognition, i-vectors, noisy speech

1. Introduction

The universal background model (UBM) has been fundamental to state-of-the-art speaker identification (SID) technology for over a decade [1]. Recently, however, we proposed a new SID framework in which a deep neural network (DNN), trained for automatic speech recognition (ASR), was used to generate posterior probabilities for a set of states in place of the Gaussians in the traditional UBM-GMM approach [2]. In combination with an i-vector/probabilistic linear discriminant analysis (PLDA) backend, the new DNN/i-vector framework offered significant improvements on SID in the context of clean telephony speech.

Our initial work [2] provided a proof-of-concept of the DNN/i-vector framework under controlled conditions (single language and channel) using the NIST speaker recognition evaluation (SRE) 2012 data set. In this study, we wish to observe how the framework copes with multiple languages and the heavy channel degradation from multiple channels exhibited in the Defense Advanced Research Projects Agency (DARPA) Robust Automatic Transcription of Speech (RATS) data set [3]. Both language and channel are interesting aspects in the new framework. The DNN training language may not match the SID language under test, and the presence of multiple channels may break the i-vector framework assumption that a single senone posterior can be modeled using a single Gaussian.

This paper extends the DNN/i-vector framework to SID under noisy conditions by first applying a convolutional neural network (CNN) instead of a DNN to improve robustness to noisy speech. This approach is motivated by ASR research in noisy conditions [4], and our successful application of the CNN/i-vector (CNNiv) framework to LID [5]. In contrast to clean speech, we show that the CNNiv framework is comparable to the traditional UBM/i-vector (UBMiv) approach when evaluated on the RATS SID task. Further experiments analyze the impact of the CNN language on SID performance and how channel distortions hinder the performance of the CNNiv framework.

This paper is organized as follows. Section 2 provides an overview of posterior extraction from CNNs and their use in the CNN/i-vector framework. Section 3 and 4 provide the experimental protocol and results.

2. Briefs of ASR/i-vector framework

In the i-vector model [6], we assume that the the following distribution generates the t-th speech frame $\mathbf{x}_t^{(i)}$ from the i-th speech sample:

$$\mathbf{x}_t^{(i)} \sim \sum_k \gamma_{kt}^{(i)} \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_k) \quad (1)$$

where the $\mathbf{T}_k$ matrices describe a low-rank subspace (called total variability subspace) by which the means of the Gaussians are adapted to a particular speech segment, $\boldsymbol{\omega}^{(i)}$ is a segment-specific standard normal-distributed latent vector, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of the k-th Gaussian, and $\gamma_{kt}^{(i)}$ is the posterior of the k-th Gaussian, given by

$$\gamma_{kt}^{(i)} = p(k|x_t^{(i)}). \quad (2)$$

Traditionally, the Gaussians in the UBM are used to define the classes k in (1). This approach ensures that the Gaussian approximation for each class is satisfied (by definition) and provides a simple way to compute the posteriors needed to compute the i-vectors. The likelihood of each Gaussian is computed and Bayes rule is used to convert them into posteriors.

In our recent work [2], we proposed the use of the classes k in (1) as the senones defined by the ASR decision tree as opposed to the Gaussian indices in a GMM. The senones are defined as states within context-dependent phones. They can be, for example, each of the three states within all triphones. They are the unit for which observation probabilities are computed during ASR. The pronunciations of all words are represented by a sequence of senones $\mathcal{Q}$. By using the set $\mathcal{Q}$ to define the

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA or its contracting agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. "A" (Approved for Public Release, Distribution Unlimited)

Copyright © 2014 ISCA

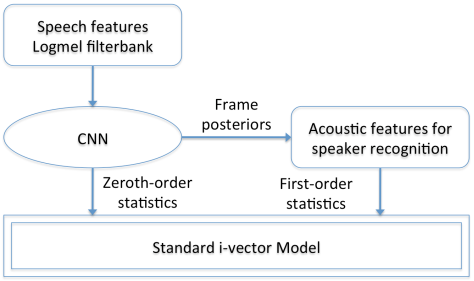

Figure 1: The flow diagram of CNN/i-vector hybrid system for i-vector model.

classes k, we make the assumption that each of these senones can be accurately modeled by a single Gaussian. While this is a strong assumption, results show that it is reasonable for the NIST SRE12 clean telephone task [2]. In that work, we used a DNN to extract the posterior probabilities for the i-vector training and extraction; however, other tools may be applicable to this task.

In this study focused on SID in noisy conditions, CNNs are used instead of DNNs for posterior probabilities extraction to enhance noise robustness. Figure 1 presents the flow diagram of the CNN/i-vector hybrid system for i-vector modeling. First, a CNN trained for ASR is used to extract the posteriors for every frame. Then, instead of the posteriors generated by the UBM in the traditional UBM/i-vector framework, the posteriors from the CNN are used to estimate the zeroth and first order statistics for the subsequent i-vector model training. Note that in this approach, we are not restricted to a single set of features for both senone posterior estimation and i-vector estimation. Indeed, the i-vector system can use features tailored more for SID than those tailored for ASR in the CNN.

2.1. CNN for Speech Recognition

For noisy conditions, CNNs were proposed to replace DNNs to improve robustness against frequency distortion in ASR. A CNN is a neural network in which the first layer is composed of a convolutional filter followed by max-pooling where the output is the maximum of the input values. The rest of the layers are similar to those of a standard DNN. CNNs were first introduced for image processing by [4, 7], and later used for speech recognition [8, 9]. In speech, the input features given to a CNN are log Mel-filterbank coefficients. Figure 2 presents an example of a convolutional layer. The target frame is generally accompanied by context information, including several filter bank feature vectors around the target frame. One or more convolutional filters are then applied to filter the feature matrix. While in image processing this filter is generally smaller than the size of the input image such that 2-D convolutions are performed, in ASR the filter is defined with the same length as the total number of frames, thus a 1-D convolution along the frequency axis is used [10]. As a result, no convolution occurs in the time domain: a single weighted sum is done across time. On the other hand, the filter is generally much shorter than the number of filter banks. This way, the output is a single vector whose components are obtained by taking a weighted sum of several rows of the input matrix.

The dimension of the output vector of the convolutional layer depends on the number of filter banks and the height of the convolutional filter. In Figure 2, there are 7 dimensional filter bank features from 5 frames (2 left, 2 right and 1 center frames) used to represent one center/target frame. The height of the convolutional filter is 2 and its width is, as mentioned above, equal

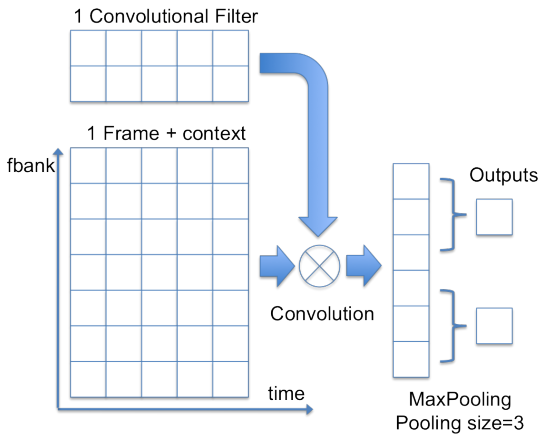

Figure 2: Diagram of a convolutional layer including convolution and max-pooling. Only one convolutional filter is shown in this example and non-overlapping pooling is used.

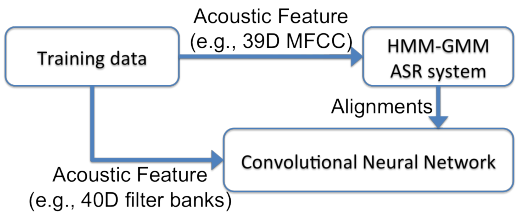

Figure 3: Flow diagram for CNN training for ASR.

to the number of frames included in the input. Since we ignore the boundary, the output of the convolution is a 6-dimensional vector.

After the convolutional filter is applied, the resulting vector goes through a process called max-pooling which selects the maximum value from N adjacent elements. This process can be done with or without overlap. The process of max-pooling is expected to reduce the distortion because it selects the largest value from a set of adjacent filter banks (which have already gone through convolutional filtering). In the example, the pooling size is 3 and no overlap is used, resulting in a 2-dimensional output.

In practice, usually 40 filter banks with a context of 15 frames are used. The height of the convolutional filter is generally 8. Furthermore, many convolutional filters are used to model the data in more detail; we use 200. The output vectors of the different filters are concatenated into a long vector that is then input to a traditional DNN. This DNN usually includes 5 to 7 hidden layers. The output layer of the DNN contains one node for each senone defined by the decision tree.

A flow diagram for CNN training in ASR is shown in Figure 3. A pre-trained hidden Markov model (HMM) ASR system with GMM states is needed to generate alignments for the subsequent CNN training. The final acoustic model is composed of the original HMM from the previous HMM-GMM system and the new CNN.

3. Experimental Protocol

Data: Data was supplied under the DARPA RATS program [3]. The training and test sets were defined in the same manner as previously described in [11], with the additional use of the two latest data collections under the program. This study focuses on the 10 second enroll, 10 second test condition (10s-10s) in

which speaker models are enrolled using 6 recordings each with 10s of nominal speech. This focus resulted in a training set consisting of 53k re-transmissions from 5899 speakers and a matched-language test set of 85k target and 5.8 million impostor trials from 305 unique speakers. Languages present in the data were Levantine Arabic, Dari, Farsi, Pushto, and Urdu.

Features: Based on our previous work on RATS SID [11], we focus on two highly complementary short-term features: PLP and PNCC. Perceptual linear prediction (PLP) features are the standard features used in speech recognition. Power-normalized cepstral coefficient (PNCC) features use a power law to design the filter bank as well as a power-based normalization instead of a logarithm [12].

Contextualization: The SRI Phase III submission for the RATS program included the novel use of rankDCT contextualization instead of traditional deltas + double deltas. This process is closely based on [13]; however, the zig-zag parsing strategy was not used. Instead, selection of coefficients was performed by first calculating the average order of coefficient values in the DCT matrix over a set of training speech frames and taking the highest ranking 85 and 100 coefficients for PLP and PNCC features, respectively. Note that the raw feature is not appended to these rankDCT features. This parsing strategy offered an improvement of approximately 5% over the zig-zag method.

Speech Activity Detection (SAD): The use of soft-SAD in the SRI submission for the RATS Phase III was also novel. Rather than detecting speech frames using a threshold on speech likelihood ratios from a speech/non-speech GMM, we utilized every frame of audio by incorporating the speech posterior in the first-order statistics. Specifically, a sigmoid function was applied to the speech/non-speech likelihood ratio which was then used to scale the posteriors from the UBM or CNN. This approach provided around 5% relative improvement in both UBMiv and CNNiv approaches over the traditional threshold-based SAD.

CNN Senone Posteriors: To extract the posterior probability of the senones, both HMM-GMM and HMM-CNN models were trained on the RATS keyword spotting (KWS) training data which contains only Levantine Arabic and Farsi. The cross-word triphone HMM-GMM ASR with 3353 senones and 200k Gaussians was trained with maximum likelihood (ML). The features used in the HMM-GMM model were 13-dimensional MFCC features (including C0), with first and second order derivatives appended. The features were pre-processed with speaker-based cepstral mean and covariance normalization (MVN). A convolutional layer followed by a DNN was trained with cross entropy using the alignments from the HMM-GMM. 200 convolutional filters and a pooling size of three were used. The subsequent DNN included five hidden layers with 1200 nodes each and an output layer with 3353 nodes representing the senones. The input feature was composed of 15 frames (7 frames on each side of the frame of interest) where each frame is represented as 40 log Mel-filterbank coefficients. The CNN was used to provide the posterior probability in the proposed framework for the 3353 senones defined by a decision tree. The training data was used to estimate $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ in (1).

I-vector Systems: We used a standard i-vector / probabilistic linear discriminant analysis (PLDA) framework as our speaker recognition system [6, 14]. Framework models were learned from the entire training set, while the 2048-component UBM was learned from a channel- and language-balanced subset of 9k segments. Results are also reported based on i-vector fusion [15] which was found to be more effective than score-level fusion in [11] for this data set. LDA dimensionality reduction from 600 to 200 was applied to PLP, PNCC and fusion i-vectors.

Table 1: Performance of CNNiv and UBMiv front ends using PNCC and PLP features evaluated on the RATS SID 10s-10s (enroll-test) condition. System fusion is indicated by '+'.

Front end$_{\text{Feature}}$	**Miss@1.5FA**	**EER**
UBMiv$_{\text{PLP}}$	33.3%	9.4%
UBMiv$_{\text{PNCC}}$	27.5%	8.1%
CNNiv$_{\text{PLP}}$	30.2%	8.5%
CNNiv$_{\text{PNCC}}$	29.5%	8.5%
UBMiv$_{\text{PLP}}$ + **UBMiv**$_{\text{PNCC}}$	23.9%	7.4%
CNNiv$_{\text{PLP}}$ + **CNNiv**$_{\text{PNCC}}$	27.5%	8.1%
CNNiv$_{\text{PLP}}$ + **UBMiv**$_{\text{PNCC}}$	20.4%	6.6%
CNNiv$_{\text{PNCC}}$ + **UBMiv**$_{\text{PNCC}}$	20.8%	6.7%

4. Results

In this section we first compare the CNN/i-vector (CNNiv) approach to the traditional UBM/i-vector (UBMiv) framework in the context of the channel-degraded RATS SID data. We then investigate the language and channel sensitivities of the CNN/i-vector approach. Throughout this section, we show the combination of systems via i-vector fusion to highlight the system complementarity.

4.1. Comparing CNN and UBM i-vector Approaches

Initial results focus on two highly complementary features, PNCC and PLP, in both the UBMiv and CNNiv frameworks. Results from these frontend+feature combinations are given in Table 1. Results show that PNCC provides superior performance to PLP in the UBMiv framework. The single-feature CNNiv systems perform comparably, but worse than UBMiv$_{\text{PNCC}}$. This suggests that differences between the features are normalized with this frontend. To strengthen this hypothesis, we present, in the bottom of Table 1, the two-way i-vector fusion of systems. Fusion of UBMiv systems with different features provides a 13% relative improvement in miss rate over UBMiv$_{\text{PNCC}}$, while the two-way CNNiv fusion offers an improvement of 7% over the best CNNiv system and only obtains the same performance as the single PNCC UBMiv system. We observe the best performance when CNNiv and UBMiv systems are combined, with relative improvements of 26% in miss rate and 18% in EER over the best single system, twice the relative improvement offered through fusion of the UBM-only front ends. Furthermore, the use of a single feature (PNCC) in both front ends was comparable to the use of different features for each front end. These results show that fusion of CNNiv and UBMiv front ends is considerably more complementary than using different features.

The impressive complementarity of UBMiv and CNNiv front ends as compared to feature complementarity can be explained by the fact that different features (representations of the signal) encode the same information in different ways, whereas the CNNiv and UBMiv front ends have deeper differences. The CNN models the way each speaker pronounces each senone, while the UBM models the overall divergence of the speaker's speech to the universal speaker's speech without knowledge of phones. Alternatively expressed, the UBM approach can be seen as a parameterization of the global PDF of the features for the speaker, while the CNN allows for the parameterization to happen at the senone level.

Table 2: Performance of CNNiv system when using various languages in CNN training. Results use PNCC features evaluated on the RATS SID 10s-10s (enroll-test) condition.

Front end$_{\text{Language}}$	**Miss@1.5FA**	**EER**
CNNiv$_{\text{fas+lev}}$	29.5%	8.5%
CNNiv$_{\text{fas}}$	29.7%	8.7%
CNNiv$_{\text{lev}}$	32.9%	9.2%
CNNiv$_{\text{fas}}$ + **CNNiv**$_{\text{lev}}$	27.4%	8.1%
CNNiv$_{\text{fas+lev}}$ + **CNNiv**$_{\text{fas}}$ + **CNNiv**$_{\text{lev}}$	26.1%	7.8%

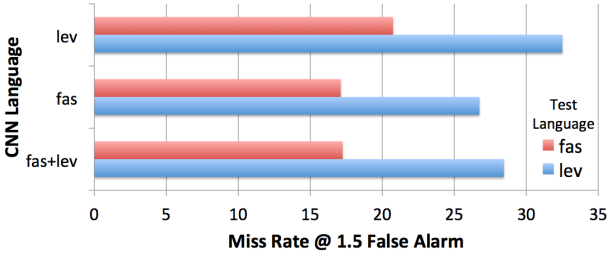

Figure 4: Illustrating the independence of CNN language and SID test language. Farsi (fas) and Levantine Arabic (lev) results are presented.

4.2. CNN Training Languages

Results in the previous section were based on a CNN trained using two languages, Farsi (fas) and Levantine Arabic (lev), via a merged dictionary. In this section, we train separate CNNs for these languages to observe the corresponding effect on CNNiv SID performance. We focus only on PNCC features. Table 2 compares CNNiv results from these three CNN models along with several fusion results. Results indicate that the dual-language CNNiv offers the best performance when evaluating matched-language tests from five languages (see Section 3) with CNNiv$_{\text{fas}}$ following closely behind. The CNNiv$_{\text{lev}}$ system offered worse performance that may be attributed to the phonetic differences of Levantine Arabic from the other four target languages which are more closely matched by the Farsi CNN. Figure 4 illustrates the performance of fas and lev tests for each model in which matching the CNN model to the test language provided no observable benefit. This lack may be due to the presence of all languages during subspace and PLDA training, but also suggests a degree of language independence in the CNNiv SID framework, despite the language-focused training of the CNN.

I-vector fusion of the single-language CNN systems provided a subtle improvement over the dual-language CNN (7% in miss rate). Additional gains found through the fusion of all three amounted to a 12% relative improvement in miss rate over the dual-language CNNiv. These gains highlight the complementarity offered through the language dependency of the CNN while providing a degree of language independence in the SID framework.

4.3. CNN/i-vector Channel Sensitivity

RATS speech data is heavily degraded by eight distinct channels. Our original DNN/i-vector framework was evaluated in clean conditions using single channel telephony speech. Here, we aimed to determine the extent to which channel variation hinders CNN performance in degraded conditions. This experiment is particularly interesting since the i-vector framework assumes that each senone posterior can be adequately modeled with a single Gaussian. We tested whether this assumption held by normalizing the first-order statistics on a channel-dependent basis. That is, a channel-specific mean and variance learned from the training set was applied to the first-order statistics extracted in the CNN/i-vector framework. We considered the case in which ground truth is known only for training data. A universal audio characterization (UAC) [16] extractor was trained using the ground truth labels. The statistics for both i-vector subspace training and i-vector extraction were normalized by the mean and variance of training segments from the *detected* channel. The i-vectors for the purpose of UAC were sourced from the UBMiv$_{\text{PNCC}}$ system and provided a average channel detection rate of 98.41% across 9 channels (including the original/clean channel)[1]. Table 3 provides results from the channel-dependent experiments. It is worth noting that all system components, including PLDA, are fully aware of channel conditions (i.e., they have observed examples of each original segment retransmitted over all eight channels) with a training set sufficient to compensate for channel effects. Nonetheless, the channel-dependent results in Table 3 provide a marginal improvement over the channel-independent system, indicating that the CNNiv system is somewhat sensitive to channel effects. One drawback to this approach to channel compensation is the need for knowledge of the UAC channel classes prior to system deployment. Table 3 also provides the performance of the PNCC-based UBMiv fused with the channel-dependent CNNiv system. The improvements from CNN-based channel compensation appear to generalize only marginally to system fusion.

Table 3: Performance of the channel-independent and channel-dependent CNNiv systems. Results use PNCC features evaluated on the RATS SID 10s-10s (enroll-test) condition. The final row shows the fusion of the UBMiv and CNNiv systems using PNCC features.

System	**Miss@1.5FA**	**EER**
Channel-independent	29.6%	8.5%
Channel-dependent	28.6%	8.3%
ChanIndep CNNiv + UBMiv	20.8%	6.7%
ChanDep CNNiv + UBMiv	20.5%	6.6%

5. Conclusions

We recently proposed the DNN/i-vector approach for SID and later proposed the CNN/i-vector framework for noise-robust language identification. In this paper, we applied the same CNN/i-vector framework to the task of SID in noisy conditions where it was found to offer performance comparable to that of the traditional UBM/i-vector framework. In fusion with a UBM/i-vector system, complementarity exceeded that of UBM-based systems using different features by an additional 13% in miss rate. The languages used to train the CNN (from an ASR framework) were found to have limited effect on SID performance of an individual system. Multiple CNNs from different languages were found to be complementary. We illustrated that channel sensitivity remains a shortcoming of the approach as yet unaddressed in unknown conditions.

[1]Use of ground truth channel labels for both training and testing speech provided SID performance on par with detected channel labels.

6. References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[2] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE ICASSP (accepted)*, 2014.

[3] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[4] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, pp. 255–258, 1995.

[5] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Speaker Odyssey Workshop (submitted)*, 2014.

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

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE ICASSP*, 2012, pp. 4277–4280.

[9] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE ICASSP*, 2013, pp. 8614–8618.

[10] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *Proc. Interspeech*, 2013, pp. 3366–3370.

[11] M. McLaren, N. Scheffer, M. Graciarena, L. Ferrer, and Y. Lei, "Improving speaker identification robustness to highly channel-degraded speech through multiple system fusion," in *Proc. IEEE ICASSP*, 2013, pp. 6773–6777.

[12] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. IEEE ICASSP*, 2012, pp. 4101–4104.

[13] M. McLaren, N. Scheffer, L. Ferrer, and Y. Lei, "Effective use of DCTs for contextualizing features for speaker recognition," in *Proc. IEEE ICASSP (accepted)*, 2014.

[14] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[15] M. Kockmann, L. Ferrer, L. Burget, and J. Cernocky, "iVector fusion of prosodic and cepstral features for speaker verification," in *Proc. Interspeech*, Florence, Italy, Aug. 2011.

[16] L. Ferrer, L. Burget, O. Plchot, and N. Scheffer, "A unified approach for audio characterization and its application to speaker recognition," in *Proc. of Odyssey Workshop*, 2012.