DEEP NEURAL NETWORK-BASED SPEAKER EMBEDDINGS FOR END-TO-END SPEAKER VERIFICATION

David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero,*
Yishay Carmiel, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
Spoken Labs, Spoken Communications, Seattle, WA 98121, USA
{david.ryan.snyder, pegahgh, dpovey}@gmail.com, {dgromero, khudanpur}@jhu.edu,
yishay.carmiel@spoken.com

ABSTRACT

In this study, we investigate an end-to-end text-independent speaker verification system. The architecture consists of a deep neural network that takes a variable length speech segment and maps it to a speaker embedding. The objective function separates same-speaker and different-speaker pairs, and is reused during verification. Similar systems have recently shown promise for text-dependent verification, but we believe that this is unexplored for the text-independent task. We show that given a large number of training speakers, the proposed system outperforms an i-vector baseline in equal error-rate (EER) and at low miss rates. Relative to the baseline, the end-to-end system reduces EER by 13% average and 29% pooled across test conditions. The fused system achieves a reduction of 32% average and 38% pooled.

Index Terms— speaker verification, deep neural networks, end-to-end training

1. INTRODUCTION

Speaker verification (SV) is the task of authenticating the claimed identity of a speaker, based on some speech signal and enrolled speaker record. Typically, a low-dimensional representation rich in speaker information is extracted for both enrollment and test speech, and is mapped to a verification score using some comparison criterion. Variants include text-dependent, where the speech content is fixed to some phrase, and text-independent SV. In this study, our interest is in real-time text-independent SV. To reduce latency, we try to minimize the amount of speech required to achieve an accurate verification. Our evaluation follows from this, and involves full-length enrollment recordings and test speech ranging from one second to a few minutes.

Speaker representations are commonly based on i-vectors [1], with a probabilistic linear discriminant (PLDA) backend used for scoring [2, 3, 4, 5, 6, 7]. Recently, this paradigm has been improved by incorporating deep neural network (DNN) acoustic models [8, 9, 10, 11, 12]. The DNN is trained for automatic speech recognition and repurposed to enhance phonetic modeling in the universal background model (UBM). Ordinarily, the components of an i-vector system are trained on complementary subtasks, but are not jointly optimized for verification. In [13], the typically generative PLDA model was trained to discriminate between same-speaker and different-speaker trials. The work in [14] went deeper into the i-vector pipeline to discriminatively train the i-vector extractor. Although our proposed system is not i-vector-based, we use a similar training criterion.

An alternative approach is to use neural networks to model speaker characteristics. Prior work includes frame-level models that compute probabilities over a fixed list of speakers [15, 16, 17, 18, 19]. After training, the output layer is discarded, and additional steps are required to aggregate frame-level representations and to perform verification. A popular approach is to train Gaussian mixture models (GMM) on bottleneck features extracted from the network. This is followed by binary classification [15, 16, 17] or i-vector extraction [19]. In [18], speaker representations are created by averaging the final hidden layer activations.

Recently, [20] introduced an end-to-end system trained to discriminate between same-speaker and different-speaker utterance pairs. This built on the frame-level approach in [18], and outperformed an i-vector baseline for a global password text-dependent SV task. Our framework is similar to the feed-forward DNN in [20], but handles variable length input through a temporal pooling layer and is developed for text-independent verification. Earlier studies in [21, 22, 23] presented a similar architecture that trains a DNN on a speaker comparison task and produces frame-level features that capture speaker characteristics. First and second order statistics are computed from these output features to create single-Gaussian speaker models. In our proposed system we also

[*] The first author was partially supported by Spoken Communications.

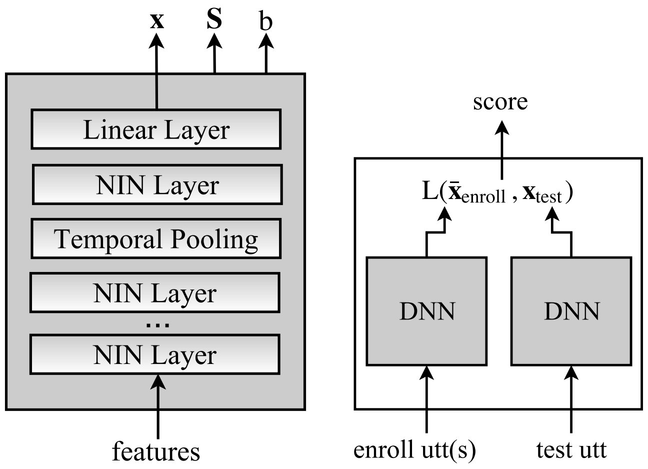
capture speaker characteristics by statistics over the utterance, but these are computed at a hidden pooling layer of the DNN and used internally. As in [20], our proposed system is trained on the same distance metric used during test-time verification.

2. BASELINE I-VECTOR SYSTEM

The baseline is a standard i-vector system that is based on the GMM-UBM Kaldi recipe described in [11]. The front-end features consist of 20 MFCCs with a frame-length of 25ms that are mean-normalized over a sliding window of up to 3 seconds. Delta and acceleration are appended to create 60 dimension feature vectors. A frame-level GMM-based VAD selects features corresponding to speech frames. The UBM is a 4096 component full-covariance GMM. The system uses a 600 dimension i-vector extractor. Prior to PLDA scoring, i-vectors are centered and length normalized. To compensate for duration mismatch, we use the strategy of truncating PLDA training data [24]. The training dataset is copied and randomly cropped to the first 1–20 seconds. The PLDA model is trained on either the short version alone, or on the combination of the short and maximum length versions.

3. DIRECT DEEP NEURAL NETWORK

3.1. Overview



(a) DNN Architecture (b) Scoring Schema

Fig. 1. Diagram of the DNN and scoring method.

The proposed architecture is a feed-forward DNN that extracts statistics over a sequence of stacked MFCCs and maps it to a speaker embedding. The objective function operates on pairs of embeddings, and maximizes a same-speaker probability for embeddings from the same speaker, and minimizes the same probability for pairs of embeddings from different speakers. Our system is built using the nnet3 neural network library in the Kaldi speech recognition toolkit [25].

3.2. Features

The features are 20 dimensional MFCCs with a frame-length of 25ms, mean-normalized over a sliding window of up to 3 seconds. 9 frames are spliced together to create a 180 dimensional input vector. After splicing, the same frame-level VAD from Section 2 filters out nonspeech frames.

3.3. Neural Network Architecture

The network, illustrated in Figure 1a, consists of four hidden layers, followed by a temporal pooling layer. The pooling layer aggregates the output of the preceding hidden layer over time and computes its average and standard deviation. These statistics are concatenated together, propagated to a final hidden layer, followed by a linear output that produces the speaker embedding $\mathbf{x}$. The symmetric matrix $\mathbf{S}$ and offset b are constant outputs (independent of the input) that are used in the distance metric, Equation 2.

The network activations are a type of network-in-network (NIN) nonlinearity recently introduced in [26]. The NIN component maps an input of dimension d_i to output of dimension d_o. Internally, a set of n micro neural networks ([27]) project the input to a d_h-dimensional space. Within a NIN component, micro neural networks share parameters, and are composed of a stack of three rectified linear units connected by affine transformations. Refer to [26] for implementation details. Our network uses the NIN configuration $\{n = 150, d_i = 600, d_h = 2000, d_o = 3000\}$, which results in a model with 6.7 million parameters.

3.4. Training

We model the probability of embeddings $\mathbf{x}$ and $\mathbf{y}$ belonging to the same speaker by the logistic function in Equation 1. Equation 2 is a PLDA-like quantity, similar to [13], that defines the distance between two embeddings. Let P_{diff} and P_{same} be the set of different-speaker and same-speaker pairs, respectively. The objective function (Equation 3) is the log probability of the correct choice for each pair. Since there are many more pairs in the set P_{diff} than in P_{same}, we introduce a constant K so that each set has the same weight in the objective function.

$$Pr(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + e^{-L(\mathbf{x},\mathbf{y})}} \tag{1}$$

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y} - \mathbf{x}^T\mathbf{S}\mathbf{x} - \mathbf{y}^T\mathbf{S}\mathbf{y} + b \tag{2}$$

$$E = -\sum_{\mathbf{x},\mathbf{y}\in P_{\text{same}}} ln\left(Pr(\mathbf{x}, \mathbf{y})\right) - K \sum_{\mathbf{x},\mathbf{y}\in P_{\text{diff}}} ln\left(1 - Pr(\mathbf{x}, \mathbf{y})\right) \tag{3}$$

Training examples are organized as pairs of same-speaker feature chunks. Minibatches are formed by picking N pairs, such that no two pairs are from the same speaker. Combining chunks across pairs results in an additional $N(N+1)$ different-speaker pairs. To handle channel variability, all chunks in the minibatch come from different recordings. Speaker embeddings are extracted from all $2N$ chunks and passed to the objective function. The derivatives with respect to $\mathbf{S}$, b and embeddings $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_{2N}$ are computed, and backpropagated to the DNN for parallelized stochastic gradient descent [28]. GPU memory limitations in conjunction with very long features (e.g., 3000 frames) restrict the size of the minibatches to $N = 10$ same-speaker pairs. Training pairs are shuffled after each iteration to ensure that many different speakers and speech cuts are compared.

In our application, it is important to minimize sensitivity to speech duration. A straightforward strategy is to train the DNN on chunks that vary widely in duration. However, we found that beginning training with the full range of chunk lengths results in unstable convergence. Our solution is to separate training into two stages: long duration chunks are presented first, followed by a mixture of short and long chunks. In the first stage, we train for two epochs using 10–30 second chunks, followed by another two epochs with 1–20 or 1–30 second chunks in the second stage.

3.5. Speaker Embeddings

Although an embedding can be extracted from a recording of any length, we found it convenient from a memory standpoint to extract embeddings from 30 second chunks and average to get an utterance-level representation. A single embedding is generated from the entire utterance if it is shorter than 30 seconds. Enrollment embeddings are extracted from one or more utterances, and averaged to create a speaker-level representation. As illustrated in Figure 1b, enroll and test utterances are scored by the distance metric used in the objective function (Equation 2).

4. DATASET

Table 1. Dataset Statistics

	#spkr	tot #rec	avg #rec/spkr	avg. dur
train5k	5k	25k	4.93	81s
train15k	15k	53k	3.53	84s
train102k	102k	226k	2.22	91s
enroll	2419	2915	1.21	91s
test	2419	2419	1	1–92s

The datasets consist of US English telephone speech. Calls are sampled at 8kHz and compressed using an internal process. Table 1 lists statistics for the datasets. The full training dataset, train102k, comprises roughly 102,000 speakers and more than 5,700 hours of speech. To explore the effect of training dataset size on performance, we found it useful to create reduced datasets train5k and train15k, with 5,000 and 15,000 speakers respectively (see Section 5.2).

The evaluation dataset consists of 2,419 speakers that do not overlap with the training speakers. Trials were constructed by randomly pairing up recordings from the evaluation speakers such that roughly 80 percent are nontarget. In total, there are 12,362 trials. Gender labels are not used, so our evaluation contains same and cross gender trials. The test conditions consist of full-length enrollment recordings compared with test segments of various lengths. Test segments are created by truncating the recordings to the first T seconds of speech, as detected by our GMM frame-level VAD. The VAD uses a threshold that corresponds to the equal error-rate on an in-domain development set. This helps to ensure that, on average, the test condition labels (e.g., 1s, 2s, etc) faithfully report the actual speech duration in the test segments. The same list of trials is used for all the duration conditions.

5. EXPERIMENTAL RESULTS

5.1. Duration Robustness

Table 2. EER(%) for Duration Robust Models

ivec102k	1s	2s	3s	5s	10s	20s	full	pool
1–20s	14.1	8.7	6.7	4.9	3.7	3.2	2.8	8.5
1–20s+full	15.0	9.4	7.0	5.1	3.8	3.1	2.6	10.0
full	16.4	9.9	7.3	5.2	3.8	2.8	2.4	10.6
dnn102k	1s	2s	3s	5s	10s	20s	full	pool
1–20s	12.6	7.5	6.0	4.2	3.4	2.6	2.5	6.0
1–30s	13.8	8.7	6.2	4.6	3.4	2.6	2.4	6.6

Our application requires high accuracy on short test segments and calibrated scores across test conditions. We therefore examine several methods to compensate for duration variability. The labels on the the first seven columns of Table 2 denote the amount of speech retained in the test segments. The final column is for pooled results. The row labels describe how the training data is configured. For i-vectors, this involves training a PLDA model on randomly truncated speech as described in Section 2. The end-to-end system uses the two stage method described in Section 3.4 to handle duration variability.

We denote i-vector and DNN systems trained on train102k as ivec102k and dnn102k respectively. For ivec102k, training the PLDA model on 1–20 second segments results in the best performance on the shortest five test conditions and pooled. Adapting dnn102k to 1–20 instead of 1–30 second chunks produces equivalent or better results on all but the full-length condition. With respect to systems using 1–20 second chunks, dnn102k outperforms ivec102k on all conditions, and achieves a relative improvement of 13% in terms of average EER and 29% in pooled EER. Since we are more concerned with the short-duration conditions, we use the 1–20 second adaptation methods for the remaining experiments.

5.2. Training Data Size

Table 3. EER(%) and Training Dataset Size

	1s	2s	3s	5s	10s	20s	full	pool
ivec5k	14.8	11.4	9.0	7.0	5.4	4.4	3.5	8.6
dnn5k	17.5	12.6	10.7	8.7	7.2	6.4	6.2	10.6
ivec15k	13.8	9.0	7.0	5.1	3.9	3.0	2.7	8.0
dnn15k	14.2	10.7	8.0	6.5	5.4	4.9	4.9	8.3
ivec102k	14.1	8.7	6.7	4.9	3.7	3.2	2.8	8.5
dnn102k	12.6	7.5	6.0	4.2	3.4	2.6	2.5	6.0

In Table 3 we report the effect of training dataset size on performance. The N in train$N$k refers to the number of speakers, in thousands. All systems use the 1–20 second duration compensation methods from Section 5.1. Trained on the smaller datasets, the i-vector system outperforms the DNN on all conditions. This is especially noticeable on the long duration conditions: ivec5k is 44% better than dnn5k on the full condition, but only 15% better on the 1s condition. However, the average performance seems to stagnate for the i-vector at 15,000 speakers, and additional speakers do not consistently improve the pooled results. On the other hand, the DNN appears better able to exploit a substantial increase in the number of speakers. The average EER improves by 21% from dnn5k to dnn15k, and 29% from dnn15k to dnn102k.

5.3. System Combination

Table 4. EER(%) of System Fusion

	1s	2s	3s	5s	10s	20s	full	pool
ivec102k	14.1	8.7	6.7	4.9	3.7	3.2	2.8	8.5
dnn102k	12.6	7.5	6.0	4.2	3.4	2.6	2.5	6.0
fusion	10.2	6.1	4.3	3.4	2.4	1.9	1.6	5.3

The DNN performs well by itself, but due to the significant architectural differences between it and the i-vector baseline, we anticipate that the systems are excellent candidates for fusion. To fuse ivec102k and dnn102k, we first normalize the scores using mean and variance calculated from all pooled scores and add them together. Relative to the baseline, the fused system is 32% and 38% better, in terms of average and pooled EER.

5.4. DET Curves

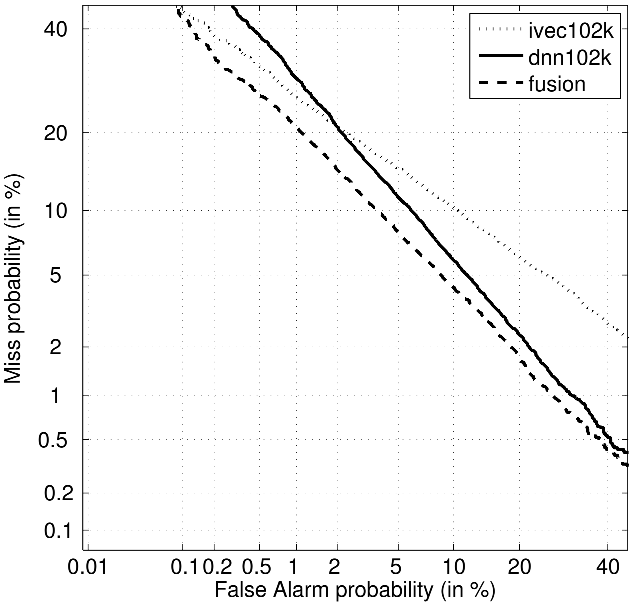


Fig. 2. DET curve for the pooled 1s, 2s, 3s, and 5s conditions.

So far, we have compared systems at the EER operating point, in which the false alarm rate (FAR) equals the miss rate (MR). However, for many verification applications, avoiding false alarms is prioritized. Therefore, we plot detection error tradeoff (DET) curves for the individual and fusion systems. Relative to the i-vector, the DNN performs better at a low MR and worse at a low FAR. Figure 2 plots a DET curve for the 1–5 second test conditions. We see that ivec102k and dnn102k overlap at 2% FAR and 20% MR. The baseline ivec102k is better for FAR less than 2%, although the DNN is better everywhere else. With the exception of extremely low FAR, the fusion system is the same or better than the individual systems. Figure 3 shows a similar pattern for the long duration test conditions, but the cross over occurs at 2% FAR and 4.5% MR. This indicates that the DNN dominates over a larger set of operating points for short duration test conditions than for long.

6. CONCLUSIONS

We studied a deep neural network architecture that extracts speaker embeddings from variable length speech segments, and scores them using the distance metric from the objective function. In [20], it was suggested that neural network-based end-to-end architectures are generally applicable to verification tasks. Our findings agree with this, and show promis-

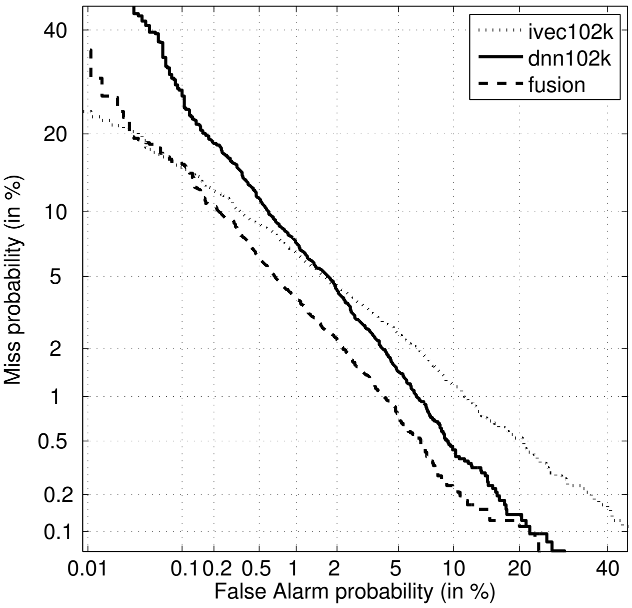


Fig. 3. DET curve for the pooled 10s, 20s, and full conditions.

ing results for text-independent speaker verification, given an adequate number of training speakers. We found that the proposed architecture outperformed our i-vector baseline by 13% average and 29% pooled EER. The larger relative improvement for the pooled error-rate and a better DET curve on the short test conditions suggest that the DNN-based embeddings may be more robust to duration variability, and better at modeling speaker characteristics from small amounts of speech. In future work, we plan to investigate speaker embeddings for ASR speaker adaptation and speaker diarization.

7. REFERENCES

[1] 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.

[2] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8.

[3] N. Brümmer and E. De Villiers, "The speaker partitioning problem.," in *Odyssey*, 2010, p. 34.

[4] J. Villalba and N. Brümmer, "Towards fully bayesian speaker recognition: Integrating out the between-speaker covariance.," in *INTERSPEECH*, 2011, pp. 505–508.

[5] P. Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

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

[7] D. Garcia-Romero, X. Zhou, and C. Espy-Wilson, "Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4257–4260.

[8] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey*, 2014.

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

[10] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 378–383.

[11] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 92–97.

[12] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1671–1675, 2015.

[13] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.

[14] O. Glembek, L. Burget, N. Brummer, O. Plchot, and P. Matejka, "Discriminatively trained i-vector extractor for speaker verification," in *Interspeech*, 2011.

[15] Y. Konig, L. Heck, .M Weintraub, and K. Sonmez, "Nonlinear discriminant feature extraction for robust text-independent speaker recognition," in *Proc. RLA2C, ESCA workshop on Speaker Recognition and its Commercial and Forensic Applications*, 1998.

[16] L. Heck, Y. Konig, K. Sonmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker

recognition by discriminative feature design," in *Speech Communication*, 2000, vol. 31, pp. 181–192.

[17] T. Yamada, L. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of dnn.," in *Interspeech*, 2013, pp. 3661–3664.

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

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

[20] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[21] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture,," in *IEEE Transactions on Neural Networks*, 2011, vol. 22, pp. 1744–1756.

[22] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," in *Advances in Neural Information Processing Systems (NIPS11)*, 2011.

[23] A. Salman, *Learning speaker-specific characteristics with deep neural architecture*, Ph.D. thesis, University of Manchestery, 2012.

[24] T. Hasan, R. Saeidi, J. Hansen, and D. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7663–7667.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*, 2011.

[26] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using cnns," in *To appear in Interspeech 2016*. IEEE, 2016.

[27] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[28] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *CoRR*, vol. abs/1410.7455, 2015.