INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances

Chunlei Zhang[1,2], *Kazuhito Koishida*[2]

[1]Center for Robust Speech Systems (CRSS), University of Texas at Dallas, Richardson, TX 75080
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

chunlei.zhang@utdallas.edu, kazukoi@microsoft.com

Abstract

Text-independent speaker verification against short utterances is still challenging despite of recent advances in the field of speaker recognition with i-vector framework. In general, to get a robust i-vector representation, a satisfying amount of data is needed in the MAP adaptation step, which is hard to meet under short duration constraint. To overcome this, we present an end-to-end system which directly learns a mapping from speech features to a compact fixed length speaker discriminative embedding where the Euclidean distance is employed for measuring similarity within trials. To learn the feature mapping, a modified Inception Net with residual block is proposed to optimize the triplet loss function. The input of our end-to-end system is a fixed length spectrogram converted from an arbitrary length utterance. Experiments show that our system consistently outperforms a conventional i-vector system on short duration speaker verification tasks. To test the limit under various duration conditions, we also demonstrate how our end-to-end system behaves with different duration from 2s-4s.

Index Terms: speaker verification, triplet loss, Inception network, short duration

1. Introduction

Speaker verification (SV), which offers a natural and flexible solution for biometric authentication, has been actively studied in the past decades. According to different application scenarios, speaker verification can be categorized into *text-dependent* and *text-independent* [1]. The text-dependent SV system requires the same set of phrases for enrollment and test. Combined with a keyword spotting system (KWS), text-dependent SV can be integrated into an intelligent personal assistant such as Apple Siri, Amazon Alexa, Google Now and Microsoft Cortana, where KWS and text-dependent SV serves as a keyword voice-authenticated wake-up to enable the following voice interaction [2, 3, 4]. Recent advancements on text-dependent SV tasks have been reported using deep neural networks (DNNs) and recurrent neural networks (RNNs) for speaker discriminative or phonetic discriminative network training, where intermediate frame-level features such as d-vectors, bottleneck activations or phonetic alignments are extracted to formulate utterance-level speaker representations [4, 5, 6]. More recently, DNNs, RNNs and convolution neural networks (CNNs) with a end-to-end loss $\log P(accept/reject)$ are investigated in the global keyword (e.g.,"OK Google" and "Hey Cortana") speaker verification tasks [2, 3], and are shown to achieve better performance compared with conventional techniques such as GMM-UBM and i-vector/PLDA.

* Chunlei Zhang performed the work while he was an intern at Microsoft Corporation, Redmond, WA.

In the context of text-independent speaker verification, i-vector/PLDA framework and its variants are still the state-of-the-art in most of the tasks [6, 7]. In recent two NIST SREs (e.g., SRE12 and SRE16) and their post-evaluations, almost all leading systems are based on i-vectors [8, 9]. However, i-vector systems are prone to have performance degradation when short utterances are met in enrollment/test phase. Fig.1 is the DET curves with respect to different durations of test utterances in CRSS submissions for SRE16 [10]. A clear speaker verification performance drop can be found in this analysis. It is not surprising because more data in the MAP adaption step always leads to more robust i-vector estimation [11].

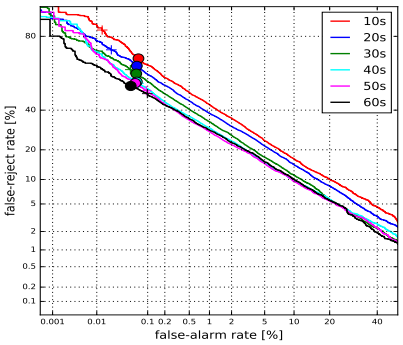

Figure 1: *i-vector based system performance versus different durations in SRE16*

To compensate for insufficient information or context mismatch due to short duration in i-vector based text-independent SV, several techniques such as replacing the UBM posteriors with more supervised phonetic DNN posteriors, introducing of uncertainty into i-vector extraction or employing subspace GMM (SGMM) and Joint Factor Analysis (JFA) to train phonetic context invariant i-vectors are proposed at the acoustic model level [6, 12, 13]. At a back-end level, length normalization and quality measure function based score calibration are reported to be effective for this problem [14, 15].

Using different deep learning frameworks with end-to-end loss functions to train speaker discriminative embeddings has drawn more attention recently. In [16], DNNs with network-in-network activations and an empirically designed loss function achieved better speaker verification performances when 105k speakers were employed in the network training. More recently, Bi-LSTMs with triplet loss function from face recognition community are reported to achieve better performance in the "same/different" speaker detection experiment compared with Bayesian Information Criterion (BIC) and Gaussian Divergence with a small scale dataset [17]. From the results in both [16, 17], it seems that end-to-end systems with speaker embedding are very promising to have better performances on short duration compared with i-vector systems.

http://dx.doi.org/10.21437/Interspeech.2017-1608

In this study, we aim to investigate on a new end-to-end system for text-independent speaker verification on short utterances. Similar to [18]. we convert each arbitrary length utterance into a fixed length spectrogram by cropping or padding, and the generated spectrogram is the input feeding into deep nets. Unlike the system proposed in [17, 19], we introduce a modified Inception network with residual connections to generate speaker embedding, which is the state-of-the-art deep learning architecture in image classification [20]. For the end-to-end objective training function, the same triplet loss as [17, 19] is employed while new constraint in triplet sampling is imposed to make the training run smoothly. In addition, our training procedure utilizes recent advancements in deep learning community such as batch normalization, network reduction to solve the difficulty of training with triplet loss [20, 21]. To measure the similarity within trials, Euclidean distance is utilized on speaker embeddings. Finally the end-to-end system is evaluated on speaker verification task with a Short Duration Corpus.

Although more detailed explanations and analysis can be found throughout this paper, let us first summarize the contributions here: a) providing a novel deep learning based method for text-independent speaker verification other than i-vector framework, especially for short utterance; b) this end-to-end approach results in considerably simplified systems requiring fewer concepts and heuristics; c) create a possibility for a lot of applications such as speaker change detection, speaker diarization and speaker adaption for speech recognition etc.

2. End-to-End speaker verification system

This section describes an overview architecture of our proposed end-to-end speaker verification, which is inspired by recent advancements in both face recognition and speaker recognition [17, 19]. The detail of its essential components and modifications for speaker embedding network training are presented in the following sections.

2.1. System structure

Fig.2 depicts the structure of our proposed end-to-end system for speaker verification. The system consists of a batch input layer and a deep architecture (can be flexible to apply many different deep nets) followed by L_2 normalization, which results in the speaker embedding for speaker verification. The L_2 Norm constrains the speaker embedding into an unit hypersphere to make the deep learning objective optimization and final similarity measure within certain realms.

Given an already defined deep architecture with parameters θ, and considering it as a black box function that maps an utterance into a feature space $\mathbb{R}^d$, such that the *distance* between paired utterances of the same speaker ID is small, in the meanwhile the *distance* between paired utterances of different speaker IDs is large for any channel and SNR conditions etc. To simplify the expression, the embedding is represented by $f_\theta(x) \in \mathbb{R}^d$. With L_2 normalization, the d-dimensional feature vector satisfies the constrain, i.e., $\|f_\theta(x)\|_2 = 1$. The network parameters θ is learned with an objective function. To achieve the objective of learning a speaker discriminative embedding from an utterance, triplet loss is employed in this study.

2.2. Triplet loss

To make an utterance x_i^a (anchor) of a specific person more similar to all other utterances x_i^p (positive) of the same person than it is to any utterance x_i^n (negative) of any other person,

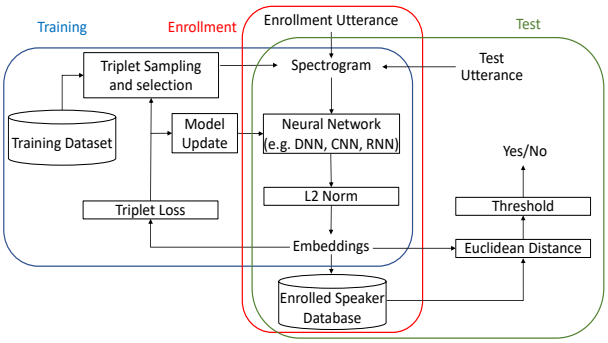

Figure 2: *The architecture of our end-to-end triplet loss based system for text-independent speaker verification.*

e.g., the training wants embeddings to follow Equation 1:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \quad (1)$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T}$$

where α is an empirically defined margin that is enforced between positive and negative pairs. $\mathcal{T}$ is the set of triplets, $(f(x_i^a), f(x_i^p), f(x_i^n))$ is a triplet. With Equation 1, the triplet loss is formulated as Equation 2 with the objective to minimize this loss over the whole set $\mathcal{T}$:

$$L = \sum_i [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha], \quad (2)$$

2.3. Triplet sampling and selection

Similar to the triplet sampling strategy proposed in [17, 19], we also have to select triplets which violate the constraint described in Equation 1. A change has been made to [17] where we always randomly select a small number of speakers from the speaker pool for each epoch. We believe this strategy will make no difference from training with a large number of epochs. At the same time, we can observe the performance on validation set to better monitor the training process.

In our experiment, we sample 60 speakers at one time, and randomly select 40 utterances for each speaker. Following the same triplet generation method as [17], we have 60x40x39/2=46800 triplets for one epoch. And we further reduce the triplet number by only selecting the ones which violate the constraint of Equation 1 with $\alpha = 0.2$.

2.4. Inception-resnet-v1 network [20]

The network architecture proposed for our end-to-end system training is Inception-resnet-v1, which is the state-of-the-art framework for image classification tasks in computer vision community. Compared with the Inception network employed in the Facenet paper [19], the Inception-resnet-v1 network achieves faster convergence without adding additional computation complexity. With the introduction of residual connections to the Inception network, the triplet loss training difficulty is alleviated. Fig.3 is a simplified diagram of Inception-resnet-v1 network, for more details about this very deep CNN based network architecture, please refer [20]. It should be noted that the Inception-resnet-v1 is hand-craft designed and only hyperparameter which needs to be tuned is the embedding size controlled by the final fully connected layer. Also, different network architectures can be applied to our end-to-end system, such as Inception network and Bi-LSTM which are already

proved to be effective in similar tasks. In this study, to better capture the global identity structure from the spectrogram feature, we utilize a very deep network architecture–Inception-resnet-v1.

As mentioned above, we extract forced aligned spectrogram from variant length speech utterances as the input to Inception-resnet-v1 network, similar to the strategy proposed in [18]. In the frequency domain, we reserve 0-5K range, and make the spectrogram with a height of 160. We set 4s as the length in our primary speaker verification experiment, with this setup, we can create a 160×250 2-d image from one utterance. To be clear, the height and length is based on 16K Hz sample-rate and 512 point FFT. We also explore the speaker discriminative capability of our end-to-end system with even shorter duration such as 3s and 2s. On these conditions, the height of spectrogram doesn't change, while the length varies according to the duration.

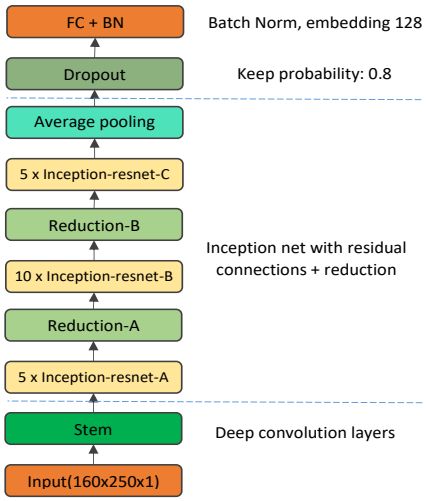

Figure 3: *A simplified architecture of Inception-resnet-v1 network*

2.5. Speaker verification evaluation

After the training of the end-to-end system, the embedding can be considered as a speaker representation and used to measure the similarity between speakers. In this phase, we use negative Euclidean distance between pairs as the likelihood score to make the decision.

$$S(x_{enroll}, x_{test}) = -\|f(x_{enroll}) - f(x_{test})\|_2^2, \quad (3)$$

With this score metric for the back-end, a pure end-to-end speaker verification system is developed. In fact, we still can utilize the back-end classifiers developed from speaker verification community, i.e., the utterance level speaker embedding can be treated as i-vector, and state-of-the-art PLDA can be applied straight forward. In our system, PLDA is not utilized since our embedding is trained in an end-to-end manner.

3. Corpus

The corpus[1] that we use for the network training, system validation and final evaluation is a large collection of speakers consisting of recordings from three different mainstream platforms, i.e., Android, iPhone and Windows Phone. To better describe this dataset, the corpus statistics are given in the next section.

[1] http://kingline.speechocean.com/exchange.php?id=1191&act=view

3.1. Corpus statistics

The corpus used in our experiments consists of about 2800 speakers where there are around 300 short utterances from each speaker. The duration distribution are illustrated in Fig.4 and the mean duration is 4s. The corpus is split into training, validation, and test with the ratio of 60%, 20%, and 20%, respectively.

Table 1: *Corpus statistics*

	Android	iPhone	WinPhone	total	mean/s
training	954	470	249	1673	4.02
validation	318	156	83	557	3.98
test	319	158	83	560	3.97

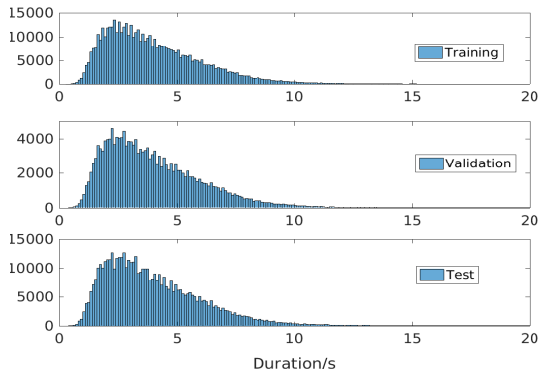

Figure 4: *Duration distributions of training, validation, test set.*

3.2. Trial list for validation and test

To validate and monitor our system training, we select 180 speakers from the validation set and create 190 target and 179 nontarget trials for each speaker. For the system performance evaluation, 450 speakers are picked from the test pool. For each speaker, 10 utterances are sampled as the enrollment data. Other than the 10 enrolled utterances, we sample 80 utterances each from the same and different speakers. In total, we create 720K trials for testing.

4. Experiments

4.1. i-vector baseline

The baseline is a standard i-vector system that is based on the GMM-UBM Kaldi SRE10 V1 [22]. The front-end features consist of 20 MFCCs with a frame-length of 30ms that are mean-normalized over a sliding window of up to 3 seconds. Delta and acceleration are appended to create 60 dimension feature vectors. Unvoiced parts of the utterances are removed with energy based voice activity detection. The UBM is a 1024 component full-covariance GMM. The system uses a 400 dimension i-vector extractor. Prior to PLDA scoring, i-vectors are centered and length normalized. We use all the training dataset for the UBM, T-Matrix and PLDA training.

4.2. Performance versus epoch number on validation set

To observe the performance of our end-to-end system on the validation set in the training, two different metrics are utilized in the experiment. The first one is validation rate (VAL) at a specified false accept rate (FAR), follow the setup in Facenet paper, "VAL@10^{-3}FAR" is adopted. The definition of VAL and FAR is given as:

$$\mathrm{VAL} = \frac{\#\mathrm{TA}}{\#\mathrm{Target}}, \mathrm{FAR} = \frac{\#\mathrm{FA}}{\#\mathrm{Nontarget}}, \quad (4)$$

where $\#\text{TA}, \#\text{FA}, \#\text{Target}, \#\text{Nontarget}$ are the number of true accept, false accept, target trial and nontarget trial respectively. By setting a threshold such that a very low FAR (i.e., 10^{-3}) is fixed, VAL will increase with the training progress.

Another metric which we use is EER. The stop criteria for training is based on these metrics. In fact, as shown in Fig.5, VAL still goes up even EER saturates after 60 epochs. In the test set, we get better performance from the model with a higher VAL although the EER is almost the same.

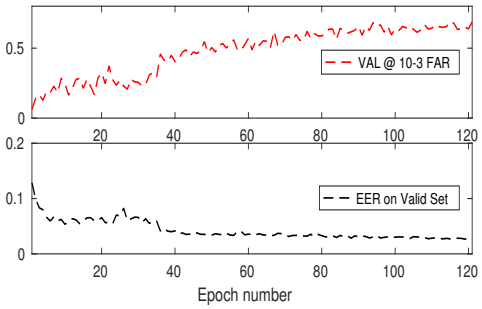

Figure 5: *The speaker verification performance across epochs. **learning rate**=0.1 is used for the first 36 epochs, 0.01 is used until 60 epochs, since than a decay rate 0.5 is applied every 20 epochs, and the RMSProp [23] optimizer is employed throughout the learning process.*

4.3. Performance on test set

In Table 2 we report the primary results of our end-to-end system on 4s condition. The best end-to-end system gives 17.0% relative improvement over the i-vector/PLDA baseline. Equal weights score fusion of "e2e 120 E" and "i-vector/PLDA" further boost the performance by 19.8% due to the significant architectural differences between our end-to-end system and the i-vector baseline.

Table 2: *Utterance level speaker verification performance on test set, results from 3 different training stages (e.g., 40, 80 and 120 epochs) are presented.*

system	e2e/40 E	e2e/80 E	e2e/120 E	i-vector/PLDA	fusion
EER	3.98%	**3.26%**	**2.97%**%	3.58%	**2.87%**

4.4. Performance against shorter duration

We present the system performance on different duration conditions in Fig.6. More specifically, 4s, 3s and even shorter 2s conditions are tested with our end-to-end framework. A DET curve for the i-vector/PLDA baseline is also illustrated for system comparison. In Fig.6, a performance degradation is observed when we cut the duration from 4s to 3s or 2s. In terms of EER, the 3s condition seems to have a comparative performance with the i-vector/PLDA system (slightly better: 3.43% VS. 3.58%). And it seems that our end-to-end system performs better on short duration condition as the mean duration of corpus is around 4s. It is also interesting to see that three end-to-end systems behave consistently: better performance in False Alarm rates while worse performance in False Reject dimension compared with the i-vector/PLDA system.

4.5. Performance by number of enrollment utterances

It should be noted that both validation and test trial lists are at utterance level. In typical speaker verification, there is always

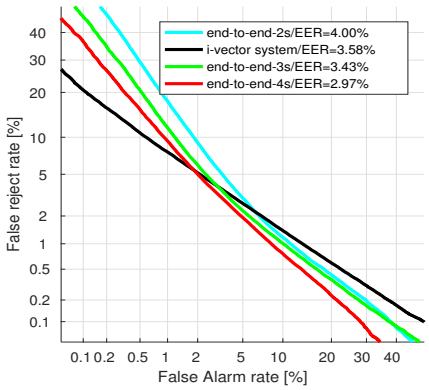

Figure 6: *DET plots on different duration conditions.*

an enrollment step using multiple enrolled utterances to build robust speaker models. In our experiments, we average scores across enrollment utterances to make a final speaker level decision. And this is an alternative option besides averaging of i-vectors/embeddings in enrollment step [24].

We present the result with different number of utterances for enrollment in Table 3. As mentioned in Sec.3.2, we perform enrollment action at score level by averaging scores from the same test utterance. More than 37% of relative improvement has been achieved when the number of enrolled utterance increases from 1 to 10. From Table 3, a relatively larger performance boost is observed when 2 and 5 utterances are enrolled, while the benefit of multiple utterances for enrollment seems to be saturate after 5 utterances. The observation can be a guidance for design of many real world speaker verification systems, especially when only limited computational resource can be utilized.

Table 3: *Speaker level speaker verification performance on test set, with 1, 2, 5, 10 utterances for enrollment in terms of EER.*

# enroll utts	1	2	5	10
i-vector/PLDA	3.58%	2.76%	2.03%	1.97%
end-to-end	**2.97%**	**2.41%**	**1.94%**	**1.84%**

5. Conclusion and future work

In this study, we present a novel end-to-end text-independent speaker verification system. Triplet loss function allows us to optimize the entire system in an end-to-end manner without introducing many concepts and heuristics. A very deep CNN based network architecture called Inception-resnet-v1 is successfully employed for the training of speaker discriminative embedding. As a result Euclidean distance is directly applied to measure similarity within trials, which is essential for our end-to-end system. It is shown from experiments that our proposed end-to-end system achieves consistently better performance over the i-vector system.

We believe that this paper shows potential of our framework for speaker recognition and related area. There are still a lot of room for improvements in our system, by different neural network architectures, loss functions, vector normalization, etc. At the same time, this approach can be directly applied to many other applications such as speaker change detection, speaker diarization, and speaker adaption for speech recognition. Other directions like language identification or emotion recognition from speech are also promising. The results here show both meaningful advancements, as well as a point to direction for future research.

6. References

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *IEEE ICASSP*, 2016.

[3] S. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," *arXiv preprint arXiv:1701.00562*, 2017.

[4] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE ICASSP*, 2014.

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

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

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

[8] "NIST speaker recognition evaluation 2012, SRE12," http://www.nist.gov/itl/iad/mig/sre12.cfm, 2012.

[9] "NIST speaker recognition evaluation 2016, SRE16," https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016, 2016.

[10] C. Zhang, F. Bahmaninezhad, S. Ranjan, C. Yu, N. Shokouhi, and J. H. Hansen, "Utd-crss systems for 2016 nist speaker recognition evaluation," in *ISCA INTERSPEECH17*, 2017.

[11] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

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

[13] T. Stafylakis, P. Kenny, V. Gupta, J. Alam, and M. Kockmann, "Compensation for phonetic nuisance variability in speaker recognition using dnns," in *ISCA Odyssey*, 2016.

[14] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *ISCA INTERSPEECH11*, 2011.

[15] T. Hasan, R. Saeidi, J. H. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *IEEE ICASSP*, 2013.

[16] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network based speaker embedding for end-to-end speaker verification," 2016.

[17] H. Bredin, "TristouNet: Triplet Loss for Speaker Turn Embedding," in *IEEE ICASSP*, 2017.

[18] C. Zhang, C. Yu, and J. H. Hansen, "An investigation of deep learning frameworks for speaker verification anti-spoofing," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR*, 2015.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE ASRU*, 2011.

[23] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp." technical report, 2012.

[24] G. Liu and J. H. Hansen, "An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1978–1992, 2014.