Interspeech 2018
2-6 September 2018, Hyderabad

Avoiding Speaker Overfitting in End-to-End DNNs using Raw Waveform for Text-Independent Speaker Verification

Jee-Weon Jung[1], *Hee-Soo Heo*[1], *IL-Ho Yang*[1], *Hye-Jin Shim*[1], *and Ha-Jin Yu*[1†]

[1]School of Computer Science, University of Seoul, South Korea

jeewon.leo.jung@gmail.com, zhasgone@naver.com, heisco@hanmail.net,
shimhyejin930615@gmail.com, hjyu@uos.ac.kr

Abstract

In this research, we propose a novel raw waveform end-to-end DNNs for text-independent speaker verification. For speaker verification, many studies utilize the speaker embedding scheme, which trains deep neural networks as speaker identifiers to extract speaker features. However, this scheme has an intrinsic limitation in which the speaker feature, trained to classify only known speakers, is required to represent the identity of unknown speakers. Owing to this mismatch, speaker embedding systems tend to well generalize towards unseen utterances from known speakers, but are overfitted to known speakers. This phenomenon is referred to as *speaker overfitting*. In this paper, we investigated regularization techniques, a multi-step training scheme, and a residual connection with pooling layers in the perspective of mitigating speaker overfitting which lead to considerable performance improvements. Technique effectiveness is evaluated using the VoxCeleb dataset, which comprises over 1,200 speakers from various uncontrolled environments. To the best of our knowledge, we are the first to verify the success of end-to-end DNNs directly using raw waveforms in text-independent scenario. It shows an equal error rate of 7.4%, which is lower than i-vector/probabilistic linear discriminant analysis and end-to-end DNNs that use spectrograms.

Index Terms: speaker overfitting, speaker embedding, raw waveform, end-to-end, speaker verification

1. Introduction

With the recent success of deep learning, studies that replace individual sub-tasks with deep neural networks (DNNs) are highly popular in various audio domains [1, 2, 3, 4, 5, 6]. This trend also applies to speaker verification. Three major sub-tasks of speaker verification (i.e., raw waveform pre-processing, speaker feature extraction, and back-end classification) are each being replaced with DNN-based approaches. We use a speaker embedding scheme that trains the DNN as a speaker identifier and use the selected hidden layer as the speaker feature [7, 8]. Studies on raw waveform processing and back-end classification have also occurred [3, 6, 9]. Individual DNNs are integrated to comprise end-to-end DNNs [1, 3, 10, 11, 12, 13].

Although DNN-based approaches have been successfully used for speaker verification, there is a difference between speaker and audio domains. In the speaker embedding scheme, there is a task mismatch between the training task: speaker identification, and the actual task: speaker verification. Because speaker identification is only conducted on predefined speakers, the speaker identifier may not be generalized towards unknown speakers in speaker verification. In this study, we address this phenomenon, which we call *speaker overfitting*, where the speaker features from the speaker embedding scheme well-represent unseen utterances from known speakers but are overfitted toward known speakers. An example of speaker overfitting is shown in Figure 1 and is further described in Section 3.

We have been building raw waveform DNNs and investigated various techniques to mitigate speaker overfitting.

- Regularization methods [14, 15, 16] and recent deep learning techniques [17, 18, 19, 20, 21]
- The multi-step training scheme [10]
- Importance of pooling, which is one of the keys to the improved performance described in Section 6

Adopting various techniques, we present a raw waveform end-to-end system, which shows better performance than both the i-vector/ probabilistic linear discriminant analysis (PLDA) system and the spectrogram end-to-end system.

The rest of this paper is organized as follows. Section 2 discusses previous works. Section 3 analyzes speaker overfitting. In Section 4, a system description is provided. Key approaches to mitigate speaker overfitting are introduced in Section 5. Section 6 describes the experiments and their results. This paper is concluded in Section 7.

2. Related Works

Past studies on raw waveform processing in DNNs, speaker embedding scheme, and end-to-end DNNs provide the three foundations for this study. Many studies have been conducted to directly process raw waveforms with DNNs [1, 3, 4, 6, 22]. Among these, the strided convolution receptive field of Collobert *et al.* [22] is used here.

A speaker embedding scheme that trains a DNN as a speaker identifier is also used in this paper [7]. In this scheme, the linear activation of the selected hidden layer is extracted as the speaker feature. Since proposed, the speaker embedding scheme has been widely used in DNN-based speaker feature extraction [8, 12, 23, 24].

End-to-end DNNs are actively being researched for many tasks [11, 25, 26]. For speaker verification, beginning with Heigold *et al.*'s work, many end-to-end DNNs have been proposed [11, 12, 13]. An end-to-end DNN that inputs raw waveforms and outputs verification results [6] is used in this paper.

3. Speaker Overfitting

For speaker verification, a speaker-embedding scheme, which extracts the speaker feature from a speaker identifiers hidden layer, is widely used. In this scheme, there exists a task mismatch between the training task, speaker identification, and the

[†] Corresponding author

10.21437/Interspeech.2018-1608

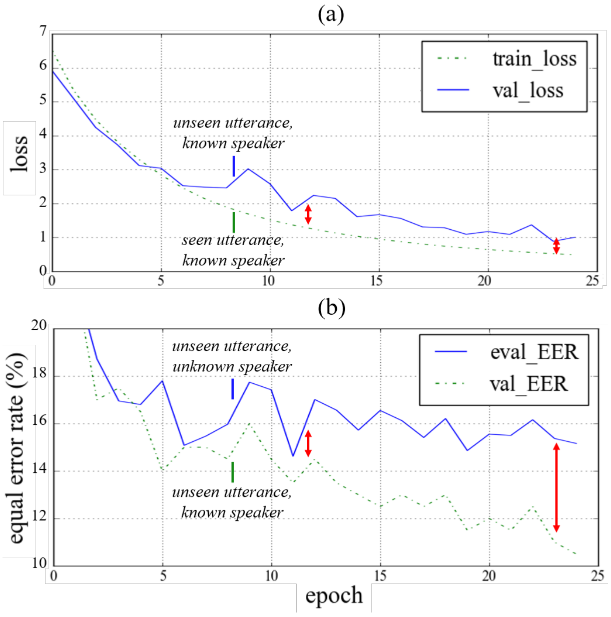

Figure 1: *(a) Loss of train (train_loss) and validation set (val_loss). (b) Equal error rates (EERs) of validation (val_EER) and evaluation set (eval_EER).*

actual task, speaker verification. Nevertheless, many successful systems using the speaker embedding scheme [7, 8, 12, 23] show that it works in a task mismatch condition.

Speaker features should be able to represent the identity of an unknown speaker for speaker verification. However, in speaker identification, all speakers are predefined and unknown speaker does not exist. Thus, speaker features in a speaker embedding scheme can be overfitted towards known speakers, which is likely to evoke performance degradation.

We call the situation where speaker features are able to represent only known speakers as speaker overfitting. We assume that it is one of the main causes of performance degradation in speaker embedding scheme.

Figure 1 depicts the result of an experiment conducted to reveal this phenomenon. Here, the dataset is divided into three subsets: train, comprising utterances of known speakers; validation, comprising unseen utterances of known speakers; and evaluation, comprising unseen utterances of unknown speakers. The train set is used to train the speaker identifier. The model is evaluated at two points. First, the generalization of unseen utterances of known speakers are evaluated using speaker identification loss in the train and validation sets (Figure 1 (a)). Second, the generalization performance of unknown speakers and the task mismatch condition are evaluated using the equal error rate (EER) of the validation and evaluation sets (Figure 1(b)). Results show that generalization of the task mismatch condition is successful, because the EER of the validation set decreases as validation loss decreases. However, generalization on unknown speakers is not successful, because the gaps between EER on the validation and evaluation sets widens. Thus, even though task mismatch does not exist in end-to-end DNNs because the training task is also speaker verification, speaker overfitting is also likely to occur.

4. System Description

In our experiment, we use raw waveform as input for analyzing speaker overfitting. Doing so, speaker verification task is operated entirely based on trainable parameters rather than human-driven techniques. This allows us to take a closer look at the effectiveness of techniques for mitigating speaker overfitting.

4.1. Speaker embedding models

Two speaker embedding models, one convolutional neural network (CNN) and one CNN-long short-term memory (LSTM) model, are exploited for our system (see Figure 2). The raw waveform CNN (RWCNN) model directly embeds the speaker feature from the raw waveform using convolutional and pooling layers. The RWCNN-LSTM model uses convolutional and pooling layers to extract a feature map, from the input raw waveform. Then, the LSTM layer, a widely used recurrent layer for processing sequential data [27, 28], is exploited to conduct sequential modeling and to embed the speaker feature. Both models extract the speaker feature from the raw waveform. However, the sequential modeling of time variation is conducted by the LSTM layer in the RWCNN-LSTM model, whereas pooling layers entirely conduct the sequential modeling in the RWCNN model. The RWCNN-LSTM model is an expansion of the RWCNN model (see Section 5.2 for details).

4.2. End-to-end model

The raw waveform end-to-end (RWE2E) model is an expanded version of the RWCNN-LSTM architecture using the b-vector scheme [9]. This model takes two raw waveforms as input and composes the b-vector via element-wise operations using two speaker features, which are those from the RWCNN-LSTM model. Element-wise operations are expected to represent the relations between the two speaker features. The b-vector is propagated through a few fully connected layers to classify whether the two utterances are from the identical speaker. The overall architecture of the RWE2E model is illustrated in Figure 3.

5. Mitigating Speaker Overfitting

5.1. Regularization

Various regularization techniques (e.g., L2 regularization and batch normalization) are key to the recent success of DNNs [16, 14, 29]. However, we argue that the importance of regularization techniques is even bigger, in terms of mitigating speaker overfitting. In the task mismatched condition, we assume that regularizing the training task improves generalization performance of the actual task. Thus, speaker overfitting is expected to be mitigated by regularizing the speaker identifier. Improvement of speaker verification performance via a simple L2 regularization during speaker identifier training also supports this claim.

5.2. Multi-step training

Deep networks often exploit pre-training schemes to show improved generalization performance. One such scheme was introduced by Heo *et al.* [10]. This scheme trains the DNN over several stages, each stage using the parameters of preceding DNN as initialization. Only the expanded layers are randomly initialized. The layers preceding the LSTM layer in the RWCNN-LSTM model are initialized using the weights of the

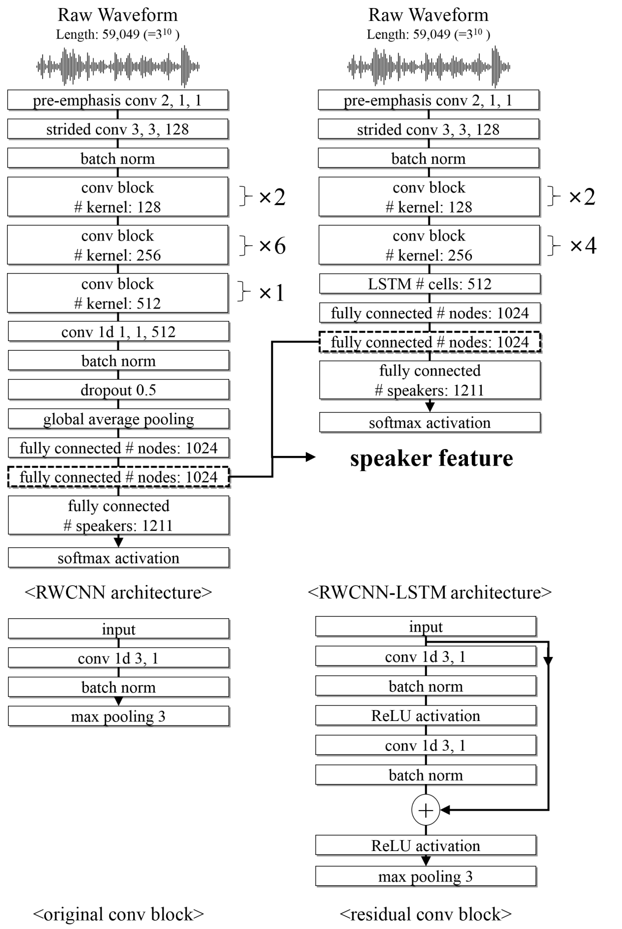

Figure 2: *Model architecture of the speaker embedding models (upper) and convolutional blocks (lower) for raw waveform models. The numbers in convolutional layers are kernel length, stride, and the number of kernels.*

RWCNN model. The RWE2E model is initialized using the weights of RWCNN-LSTM model in the same way. This step by step training scheme is called multi-step training. In Heo *et al.*'s work, multi-step training was used for fast convergence of end-to-end DNNs. In this paper, multi-step training is used to effectively mitigate speaker overfitting. Empirical results, shown in Section 6.3 support the notion that multi-step training is a key to mitigating speaker overfitting.

5.3. Residual connection and pooling layers

Residual connection [17, 18] is a recently proposed technique for training very deep architectures, showing better generalization performance in many prior studies. With residual connections, hidden layers can learn residual functions with reference to inputs. A typical residual block can be written as Equation 1, where x and y are the input and the output, respectively, of the block. W refers to the weights of hidden layers within the block, and $F(x, W)$ is the residual function.

$$y = F(x, W) + x \qquad (1)$$

Pooling layers are typically replaced with convolutional layers in DNNs having residual connections. Studies such as Springenberg *et al.*'s work [30] show that replacing pooling layers with big stride convolutional layers can improve performance of DNNs. However, in terms of mitigating speaker overfitting, usage of pooling layers is hypothesized to be one of the keys.

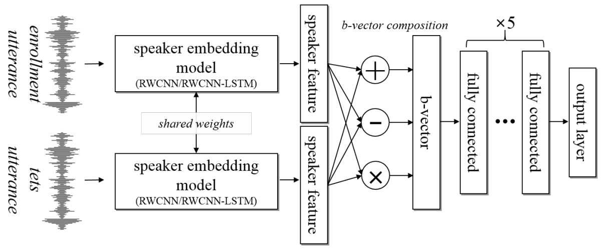

Figure 3: *Overall illustration of the raw waveform end-to-end model (RWE2E).*

In case of task mismatch condition where speaker overfitting occurs, pooling layers can be more effective in generalization because pooling layers merely reduce the information with fixed weights whereas convolution layers are trained using the training data. As the training continues, the convolutional layers are trained to identify the given speakers better and therefore can be over trained.

The pooling layer, on the contrary, can reduce the feature map size independently of the train set. Additionally, in signal processing, max pooling can be an upper envelope function for conducting smoother down-sampling, which is expected to show better generalization performance [31]. The pooling layer can also mitigate speaker overfitting by significantly reducing the number of parameters [29].

6. Experiments

6.1. Dataset

We used the VoxCeleb [13] dataset for speaker verification experiments. VoxCeleb is a public dataset for speaker recognition, comprising 1,211 speakers ($\approx$ 320 hours) as the train set, and 40 speakers ($\approx$ 10 hours) as the evaluation set. Thus, raw waveform end-to-end DNNs were explored in a text-independent scenario. Dataset partition and trial composition is identical to the Voxceleb's guideline, which makes our system performance directly comparable to [13], as shown in Table 3.

6.2. Experimental settings

All systems used raw waveforms of length 59,049 (=3^{10}) ($\approx$3.69 s) as input. Pre-emphasis embedding, an implementation of pre-emphasis using convolutional layer that has one kernel of length 2, and a strided convolution layer was used in all systems. The two parameters of pre-emphasis embedding were initialized as -0.97 and 1, and the strided convolutional layer had both a kernel length and a stride of 3.

Stochastic gradient descent was used as an optimizer with a learning rate of 10^{-3} and 0.9 momentum. L2 regularization of 10^{-4} was used. A dropout [15] rate of 50% was used only in the RWCNN model after the global average pooling layer. Batch normalization was applied in every layer in every model. The RWCNN and RWCNN-LSTM model used cosine similarity scoring as the back-end classifier.

The RWCNN models were composed of nine convolution blocks and two fully connected layers. The RWCNN model used “original conv block” as the convolution block, and the residual RWCNN model used “residual conv block”. Comparisons of each techniques effectiveness, including L2 regularization and residual connection, were made on the RWCNN model. In the RWCNN-LSTM model, one LSTM layer with 512 cells was used, followed by two fully connected layers with 1,024 nodes and an output layer. The RWCNN-LSTM models were trained using two initialization methods to compare the effectiveness of multi-step training. One used parameters of RWCNN and the other used random initialization.

In the RWE2E model, a 3,072-dimensional b-vector was composed using element-wise addition, subtraction, and multiplication of two 1,024-dimensional speaker features. Five fully connected hidden layers with 1,024 nodes were used. The output layer had two nodes, each indicating whether two utterances were from the same speaker. RWE2E models were also trained using two initialization methods: one using parameters of the RWCNN-LSTM and another using random initialization.

6.3. Results

Effectiveness of regularization techniques, multi-step training, and residual connection with pooling are described from the perspective of speaker overfitting. The RWE2E-residual model, which includes all techniques for mitigating speaker overfitting, is compared with other state-of-the-art systems. In the tables, “SID ACC” refers to accuracy of speaker identification of the validation set, and “SV EER” refers to EER of speaker verification. A technique is judged effective for mitigating speaker overfitting when the performance of speaker verification has improved, especially when the improvement occurs without the corresponding performance improvement of speaker identification on the validation set.

L2 regularization (i.e. weight decay) [16] helped mitigate speaker overfitting. By simply adopting weight decay to all hidden layers, 20% relative performance improvement was attained. Results are shown in Table 1.

Effectiveness of the multi-step training is shown in Table 2. In the RWCNN-LSTM model, multi-step training decreased speaker identification accuracy on the validation set, whereas speaker verification performance was improved. It shows that multi-step training helped mitigate speaker overfitting in a task-mismatch condition. Multi-step training also mitigated speaker overfitting in end-to-end DNNs.

Experimental results of the residual connection and pooling layers are shown in Tables 1 and 2. Table 1 shows that applying residual connections without pooling decreased performance. The residual connection with pooling layers successfully mitigated speaker overfitting, supporting our assumption from Section 5.3. Additionally, the “Inception-res-v2 model”, which shows state-of-the-art performance in image recognition [21], were tested, but did not appear to be effective in raw waveform models.

Systems performance in this paper is directly comparable with the results in [13], because the dataset configuration and trials are identical. Results are compared in Table 3, which shows that our proposed RWE2E model with L2 regularization, residual connection with pooling, and multi-step training, outperforms both the i-vector/PLDA system and the end-to-end system that takes a spectrogram as input.

Table 1: *Effectiveness of various methods in RWCNN model for mitigating speaker overfitting*

System	SID ACC	SV EER
RWCNN-w/o weight decay	88.8	14.9
RWCNN-w weight decay	90.0	12.3
RWCNN-residual-w/o pooling	78.6	18.9
RWCNN-residual-w pooling	94.1	**10.0**
RWCNN-inception-resnet-v2	94.3	11.7

Table 2: *Effectiveness of the multi-step training*

System	SID ACC	SV EER
RWCNN-LSTM-w/o multi-step	96.1	11.8
RWCNN-LSTM-w multi-step	94.6	9.2
RWCNN-LSTM-residual (proposed)	98.3	**8.7**
RWE2E-w/o multi-step	-	15.7
RWE2E-w multi-step	-	8.8
RWE2E-residual (proposed)	-	**7.4**

Table 3: *Comparison of the proposed residual-RWE2E model with other state-of-the-art systems*

System	SV EER
i-vector/PLDA[13]	8.8
spectrogram-E2E[13]	7.8
RWE2E-residual (RWCNN-LSTM init)	**7.4**

7. Conclusion and Future Works

In this paper, we explained a phenomenon we defined as speaker overfitting, in which speaker features extracted from embedding models are overfitted toward speakers within the train set. Successful adoption of residual connections was made by utilizing pooling layers, which are often replaced in residual networks. Other techniques were examined, in terms of mitigating speaker overfitting, leading to considerable performance improvement. Additionally, for the first time, a raw waveform end-to-end DNN was verified to be valid in a text-independent scenario. Furthermore, the proposed raw waveform end-to-end DNN showed better performance than both i-vector/PLDA and spectrogram-based end-to-end DNNs using the VoxCeleb dataset.

Nevertheless, direct fundamental solutions such as altering the objective function or new schemes for eliminating speaker overfitting have not been discovered yet. Our future works will be dedicated to finding these solutions.

8. Acknowledgements

This work was supported by the Technology Innovation Program (10076583, Development of free-running speech recognition technologies for embedded robot system) funded By the Ministry of Trade, Industry Energy(MOTIE, Korea)

9. References

[1] D. Palaz, M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4295–4299.

[2] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4624–4628.

[3] J. Lee, J. Park, K. Kim, Luke, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *arXiv preprint arXiv:1703.01789*, 2017.

[4] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] H. Dinkel, N. Chen, Y. Qian, and K. Yu, "End-to-end spoofing detection with raw waveform cldnns," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4860–4864.

[6] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (to be appeared)*. IEEE, 2018.

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

[8] D.Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[9] H. S. Lee, Y. Tso, Y. F. Chang, H. M. Wang, and S. K. Jeng, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1660–1664.

[10] H. S. Heo, J. W. Jung, I. H. Yang, S. H. Yoon, and H. J. Yu, "Joint training of expanded end-to-end DNN for text-dependent speaker verification," *Proc. Interspeech 2017*, pp. 1532–1536, 2017.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[12] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[13] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[14] I. Sergey and S. Christian, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[16] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in neural information processing systems(NIPS)*, 1992.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Identity mappings in deep residual networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 630–645.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, "Going deeper with convolutions." Cvpr, 2015.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[22] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[23] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–15, October 2015.

[24] T. Fu, Y. Qian, Y. Liu, and K. Yu, "Tandem deep features for text-dependent speaker verification," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[25] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[26] 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.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[30] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[31] C. R. Johnson, W. A. Sethares, and A. G. Klein, *Software receiver design: build your own digital communication system in five easy steps*. Cambridge University Press, 2011.