

Improved RawNet with Feature Map Scaling for Text-independent Speaker Verification using Raw Waveforms

Jee-weon Jung*, Seung-bin Kim*, Hye-jin Shim, Ju-ho Kim, Ha-Jin Yu†

School of Computer Science, University of Seoul, Republic of Korea

jeewon.leo.jung@gmail.com, kimholwq@naver.com, hjyu@uos.ac.kr

Abstract

Recent advances in deep learning have facilitated the design of speaker verification systems that directly input raw waveforms. For example, RawNet [1] extracts speaker embeddings from raw waveforms, which simplifies the process pipeline and demonstrates competitive performance. In this study, we improve RawNet by scaling feature maps using various methods. The proposed mechanism utilizes a scale vector that adopts a sigmoid non-linear function. It refers to a vector with dimensionality equal to the number of filters in a given feature map. Using a scale vector, we propose to scale the feature map multiplicatively, additively, or both. In addition, we investigate replacing the first convolution layer with the sinc-convolution layer of SincNet. Experiments performed on the VoxCeleb1 evaluation dataset demonstrate the effectiveness of the proposed methods, and the best performing system reduces the equal error rate by half compared to the original RawNet. Expanded evaluation results obtained using the VoxCeleb1-E and VoxCeleb-H protocols marginally outperform existing state-of-the-art systems.

Index Terms: raw waveform, speaker verification, text-independent, attention, deep neural networks

1. Introduction

With the recent advances in deep learning, many speaker verification studies have replaced the acoustic feature extraction process with deep neural networks (DNNs) [1–4]. In the preliminary stage of utilizing DNNs for speaker verification, acoustic features, e.g., Mel-frequency cepstral coefficients and Mel-filterbank energy features, were utilized as input to DNNs [5–8]. In contrast, many recent studies have also used less processed features, e.g., spectrograms and raw waveforms [9–12], hypothesizing that the usage of such less processed features as input allows data-driven approaches with DNNs to yield better discriminative representations compared to using knowledge-based acoustic features. Following this trend, many recent systems, e.g., RawNet [1], have demonstrated competitive results using raw waveforms as input for speaker verification.

The attention mechanism was initially designed to emphasize more important elements in sequence-to-sequence processing [13–17]. It has been adopted to several tasks, including speaker verification. Among various methods, self-attentive pooling has been applied to speaker verification to aggregate frame-level representations into a single utterance-level representation [16]. Here, the term “self” refers to a property of the attention mechanism that no external data, e.g., phoneme labels [18], are used. Compared to conventional global average pooling, self-attentive pooling involves assigning a weight to

each frame and conducting weighted summation. Recent attention mechanisms, e.g., multi-head self-attentive pooling, have also been investigated [17], and such methods have demonstrated further performance improvements. However, applying an attention mechanism to speaker verification is more focused on attentive pooling than applying such mechanisms to intermediate feature maps in image domain tasks [7, 16, 17].

In this study, we propose to scale the filter axis of feature maps using a sigmoid-based mechanism, which we refer to as feature map scaling (FMS). The FMS uses a scale vector whose dimension is identical to the number of filters, where each value is between 0 and 1, similar to an attention map used for an attention mechanism, with the exception that a sigmoid activation function is employed rather than a softmax function. The underlying hypothesis of using sigmoid activation functions (rather than the softmax function) to independently perform scaling is that, differing from few other tasks, an attention mechanism that exclusively selects only a few filters may remove an excessive amount of discriminative information. In addition, in light of the recent successes of attentive pooling mechanisms in speaker verification tasks, we investigate replacing the gated recurrent unit (GRU) layer of RawNet, which performs aggregation of frame-level representations, with the self-attentive pooling and self-multi-head-attentive pooling mechanisms.

Specifically, we propose to apply the FMS to feature maps by either multiplying, adding, or performing both sequentially, as shown in Figure 1. By multiplicatively scaling a feature map, we expect to emphasize each filter of a feature map independently. In addition, by applying an FMS through adding, we expect to provide small perturbations that lead to increased discriminative power. This is inspired by a previous study [19] that showed analyzing small alterations in high-dimensional space can drastically change discriminative power. By hypothesizing that these two approaches function in a complementary manner, we also propose to apply both approaches in sequence. In experiments, the proposed methods were applied to the output feature maps of each residual block following the literature [20, 21]. In addition, we investigated replacing RawNet’s first convolutional layer with a sinc-convolution layer [4], which has been reported to better capture aggregated frequency responses than the conventional convolutional layer.

The remainder of this paper is organized as follows. Section 2 describes the RawNet system, which we use as a baseline with several modifications. In Section 3, we introduce the proposed FMS. Section 4 discusses experimentation and presents an analysis of the experimental results. Finally, conclusions are presented in Section 5.

2. Baseline composition: RawNet

RawNet is a neural speaker embedding extractor that inputs raw waveforms directly without preprocessing techniques and out-

*Equal contribution

†Corresponding author

Table 1: DNN architecture of the proposed System (referred to as RawNet2 for brevity). An output layer conducts speaker identification in the training phase, and is removed after the training. BN and LeakyReLU at the beginning of the first block are omitted following [22]. Numbers denoted in Conv and Sinc-conv refers to filter length, stride, and number of filters. This architecture can also be used to represent the baseline (“#3-ours” in Table 2) by removing FMS operations and using a convolutional layer instead of a sinc-conv layer.

Layer	Input:59049 samples	Output shape
Sinc-conv	Sinc(251,1,128) MaxPool(3) BN LeakyReLU	(19683, 128)
Res block	$\left\{ \begin{array}{c} \text{BN} \\ \text{LeakyReLU} \\ \text{Conv}(3,1,128) \\ \text{BN} \\ \text{LeakyReLU} \\ \text{Conv}(3,1,128) \\ \text{MaxPool}(3) \\ \text{FMS} \end{array} \right\} \times 6$	(27, 256)
GRU	GRU(1024)	(1024,)
Speaker embedding	FC(1024)	(1024,)

puts speaker embeddings designed for speaker verification [1]. The underlying assumption behind using a DNN is that speaker embeddings extracted directly from raw waveforms by replacing an acoustic feature extraction with more hidden layers are expected to yield more discriminative representations as the amount of available data increases. RawNet adopts a convolutional neural network-gated recurrent unit (CNN-GRU) architecture, in which the first CNN layer has stride size identical to the filter length. The front CNN layers comprise residual blocks followed by a max-pooling layer, and extracts frame-level representations. Then a uni-directional GRU layer aggregates frame-level features into an utterance-level representation, which is the final timestep of the GRU’s output. The GRU layer is then connected to a fully-connected layer, where its output is used as the speaker embedding. Finally, the output layer receives a speaker embedding and performs identification in the training phase.

To construct the baseline used in this study, we implemented several modifications to the original RawNet. First, we modified the structure of the residual blocks to a pre-activation structure [22]. Second, we simplified the loss functions from using categorical cross-entropy (CCE), center [23], and speaker basis loss [24] to using only CCE loss based on empirical experiments. Third, we omitted a CNN pretraining scheme. Fourth, we modified the training dataset from VoxCeleb1 to VoxCeleb2 to utilize recently expanded evaluation protocols that consider the entire VoxCeleb1 dataset. Finally, we applied a test time augmentation (TTA) method in the evaluation phase [25] to extract multiple speaker embeddings from a single utterance by cropping with overlaps where the duration is identical to that in the training phase. Then, the average of the speaker embeddings is used as the final speaker embedding. Through these modifications, we achieve a relative error reduction (RER) of

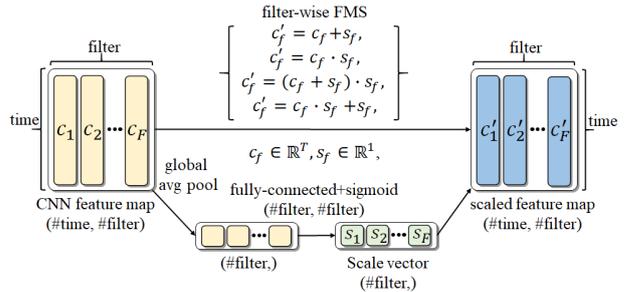


Figure 1: Illustration of the four methods using the proposed FMS. Here, s_f is broadcasted to perform element-wise calculations with c_f .

37.50 %. Throughout this study, we use cosine similarity for back-end classification to measure the improvement in speaker embedding, because various back-end classifiers have been already explored in [1].

3. Filter-wise feature map scaling

We propose to independently scale each filter of a feature map using a filter-wise feature map scaling (FMS) technique. The FMS uses a scale vector whose dimension is identical to the number of filters with values between 0 and 1 derived using sigmoid activation. Its purpose is to independently modify the scale of a given feature map (i.e., the output of a residual block) to derive more discriminative representations. We also propose various methods to utilize the FMS to scale given feature maps (i.e., multiplication, addition, and applying both). Note that these proposed approaches do not require additional hyperparameters.

Here, let $c = [c_1, c_2, \dots, c_F]$ be a feature map of a residual block, $c_f \in \mathbb{R}^T$, where T is the sequence length in time, and F is the number of filters. We derive a scale vector to conduct FMS by first performing global average pooling on the time axis, and then feed-forwarding through a fully-connected layer followed by sigmoid activation. By expressing a scale vector as $s = [s_1, s_2, \dots, s_F]$, $s_f \in \mathbb{R}^1$, we first propose to derive a scaled feature map $c' = [c'_1, c'_2, \dots, c'_F]$, $c'_f \in \mathbb{R}^T$, to scale the feature map in an additive method expressed as follows:

$$c'_f = c_f + s_f, \quad (1)$$

where s_f is broadcasted to perform element-wise calculation. We also propose to scale the feature map in a multiplicative method:

$$c'_f = c_f \cdot s_f. \quad (2)$$

These two methods can be applied sequentially, where either method can be performed first, expressed as follows:

$$c'_f = (c_f + s_f) \cdot s_f, \quad (3)$$

$$c'_f = c_f \cdot s_f + s_f. \quad (4)$$

We also propose to use two individual scale vectors, one for addition, and the other for multiplication, for (4), because it can be interpreted as $c'_f = (c_f + 1) \times s_f$. Figure 1 shows the proposed methods using FMS to scale feature maps. Here, we applied the propose methods using the FMS to the outputs of residual blocks in the baseline system following the literature [20,21].

Table 2: Performance comparison according to modifications of the baseline construction (*: data augmentation). Equal error rate (EER) is reported using the original VoxCeleb1 evaluation dataset. Ours shows the results of applying identity mapping [22], modifying the dimensionality of the code representation, and increasing the training set.

System	Trained on	TTA	EER	RER
i-vector [26]	Vox1	×	5.40	-
specCNN* [9]	Vox1	×	4.30	-
x-vector* [6]	Vox2	×	3.10	-
#1-RawNet [1]	Vox1	×	4.80	-
#2-Ours	Vox2	×	3.52	26.67
#3-Ours	Vox2	○	3.00	37.50

The proposed method using multiplicative FMS for scaling has commonality with the widely used attention mechanism [13–15] applied in the filter domain, which exclusively emphasizes a given feature map using a softmax activation. This can be interpreted as using the recently proposed multi-head attention mechanism [14] in the filter domain, where the number of the heads is equal to the number of filters. We apply scaling using a sigmoid function rather than exclusively performing scaling using a softmax function because information might be removed excessively when a conventional softmax-based attention mechanism. In translation or image classification tasks, performing exclusive concentration is reasonable; however, we hypothesize that different filters would yield complementary features for speaker verification, making independent scaling more adequate.

The proposed method with additive FMS for filter-wise scaling adds a value between 0 and 1 to a given feature map. The purpose is to apply data-driven perturbation to a feature map with a relatively small value. Here, it is assumed that this may increase the discriminative power of the feature maps. This concept is inspired by a phenomenon demonstrated by Zhang *et al.* [19], where the discriminative power of a DNN’s high-dimensional intermediate representation can differ significantly with small perturbations. In addition, we assume that applying an additive FMS combined with a multiplicative FMS will lead to further improvements.

We also investigated replacing the RawNet’s first convolutional layer with a sinc-convolution (sinc-conv) layer, which was first proposed to process raw waveforms by performing time-domain convolutions [4, 12]. It is a type of a bandpass filter, where cut-off frequencies are set as parameters that are optimized with other DNN parameters. With fewer parameters, $2 \times \#filter$, the sinc-conv layer is frequently employed in DNNs that directly input a raw waveform. Table 1 details the overall architecture of the proposed system.

4. Experiments and result analysis

All experiments reported in this paper were conducted using PyTorch [28], and the code is available at <https://github.com/Jungjee/RawNet>.

4.1. Dataset

We used the VoxCeleb2 dataset [25] for training, and we used the VoxCeleb1 dataset [29] to evaluate various protocols. The VoxCeleb2 dataset contains over one million utterances from 6112 speakers, and the VoxCeleb1 dataset contains approxi-

Table 3: Various applications of the proposed FMS. Baseline refers to the modified RawNet (Table 2). Mechanism addresses variations of applying the proposed method. c is the feature map, and s is the scale vector used to conduct FMS derived from c . “sep” indicates using **separate** scale vectors for additive and multiplicative scaling. Performance is reported in terms of EER and RER.

System	Mechanism	EER	RER
Baseline	-	3.00	-
#4-att	-	2.89	3.67
#5-multi-att	-	3.42	-
#6-SE	-	2.65	11.67
#7-CBAM	-	2.89	3.67
#8-add	$c + r$	2.82	6.00
#9-mul	$c \times r$	2.66	11.33
#10-add-mul	$(c + r) \times r$	2.60	13.33
#11-mul-add	$c \times r + r$	2.56	14.67
#12-mul-add-sep	$c \times r_1 + r_2$	2.57	14.33

Table 4: Experimental results of replacing the first strided convolution layer with varying sinc-conv layer length proposed in SincNet [4, 12]. Applied to System #11-mul-add of Table 3.

System	Sinc-conv length	EER	RER
#11-mul-add	-	2.56	-
#13	125	2.53	1.17
#14	195	2.54	0.78
#15-RawNet2	251	2.48	3.12
#16	313	2.70	-
#17	375	2.75	-

mately 330 hours of recordings from 1251 speakers for text-independent scenarios. Both datasets were obtained automatically from YouTube. Note that the VoxCeleb2 dataset is an extended version of the VoxCeleb1 dataset.

4.2. Experimental configurations

We used raw waveforms with pre-emphasis applied as input to the DNNs [1, 3, 11]. For the experiments in which the first convolutional layer was replaced with sinc-conv layer, we followed the literature [4]. This study did not apply pre-emphasis but performed layer normalization [30] to raw waveform. Here, we modified the duration of the input waveforms to 59049 samples (≈ 3.69 s) in the training phase to construction mini-batches. In the testing phasing, we applied TTA with a 20 % overlap.

We used Leaky ReLU activation functions [31] with a negative slope of 0.3 following [32]. The dimension of the speaker embedding is 1024. The AMSGrad optimizer [33] with a learning rate of 0.001 was used. A weight decay with $\lambda = 1e^{-4}$ was applied. We used CCE loss as the objective function. The other parameters related to the system architecture are described in Table 1 and the literature [1].

4.3. Results analysis

Table 2 shows the performance according to the modifications made to the RawNet system (Section 2) using the original VoxCeleb1 evaluation set. Here, the top three rows describe existing systems using the same dataset for comparison. The results

Table 5: Results of comparison to state-of-the-art systems on expanded VoxCeleb1-E and VoxCeleb-H evaluation protocols.

	Input Feature	Front-end	Aggregation	Loss	Dims	EER (%)
VoxCeleb1-E						
Chung <i>et. al.</i> [25]	Spectrogram	ResNet-50	TAP	Softmax+Contrastive	512	4.42
Xie <i>et. al.</i> [27]	Spectrogram	Thin ResNet-34	GhostVLAD	Softmax	512	3.13
Nagrani <i>et al.</i> [10]	Spectrogram	Thin-ResNet-34	GhostVLAD	Softmax	512	2.95
Ours	Raw waveform	RawNet2	GRU	Softmax	1024	2.57
VoxCeleb1-H						
Chung <i>et. al.</i> [25]	Spectrogram	ResNet-50	TAP	Softmax+Contrastive	512	7.33
Xie <i>et. al.</i> [27]	Spectrogram	Thin ResNet-34	GhostVLAD	Softmax	512	5.06
Nagrani <i>et al.</i> [10]	Spectrogram	Thin-ResNet-34	GhostVLAD	Softmax	512	4.93
Ours	Raw waveform	RawNet2	GRU	Softmax	1024	4.89

indicate that the original RawNet demonstrates competitive performance. For x-vectors, we show the results for an improved version reported in the literature [10]. Here, System #1 describes the performance of RawNet [1], and System #2 shows the result of changing the DNN architecture and expanding the training set to the VoxCeleb2 dataset from System #1. System #3 shows the results obtained by applying TTA to System #2. As can be seen, the results demonstrate that the applied changes were effective and resulted in RER of 37.5 % compared to the original RawNet. Note that we used System #3 as the baseline in all experiments.

Systems #4 through #7 of Table 3 show the results obtained by applying various related methods. Systems #4 and #5 show the results obtained using the attention and multi-head attention mechanisms using the softmax-based exclusive attention map on the filter domain. Attention demonstrated marginal improvement, and multi-head attention reduced the performance matching the hypothesis discussed in Section 3. Systems #6 and #7 describe the results of applying squeeze-excitation [21] and convolutional block attention module [20] to the baseline. In addition, the experimental results obtained by replacing the GRU layer with self-attentive pooling or self-multi-head-attentive pooling reduced performance. These results demonstrate that, in the case of RawNet, the GRU better aggregates frame-level representations into an utterance-level representation. Among application of various related methods, System #6 demonstrated the best result.

Systems #8 through #12 of Table 3 show the results of proposed FMS method with different configurations. Here, the ‘‘Mechanism’’ column shows how we performed the proposed FMS method. System #8 and System #9 applied the two proposed methods, and #10 and #11 applied both methods at the same time in different order. The results show that the proposed methods yielded improvements with RERs of 6.00 % and 11.33 %. Applying both methods simultaneously further improved the performance, and System #11 demonstrated an EER of 2.56 %. System #12 shows the result obtained using separate scale vectors for additive and multiplicative FMS. As shown, additional improvements were not observed.

Table 4 shows the results obtained by replacing the RawNet’s first convolutional layer with the sinc-conv layer of SincNet. Here, we used System #11 to perform these comparative experiments. The result demonstrates that it provides 3.12 % additional improvement in System #15. However, the performance was easily affected by the length of the sinc-conv layer, where setting an overly long filter length reduced performance. In the following, we refer to System #15 that demonstrates the

best performance as ‘RawNet2’ for brevity. RawNet2 demonstrates an RER of 48.33 % compared to the original RawNet (System #1), thereby nearly halving the EER.

Finally, Table 5 compares the results obtained in various recent studies using the expanded evaluation protocols, VoxCeleb1-E and VoxCeleb1-H, which utilize more than 1000 speakers and 500000 trials compared to 40 speakers and 38000 trials in the original evaluation protocol.¹ The results show that the proposed RawNet2 marginally outperformed the state-of-the-art performance, i.e., EER of 2.87 % for the VoxCeleb-E protocol and 4.89 % for the VoxCeleb-H protocol. From the various experimental results given through Tables 2 to 5, we conclude that the proposed RawNet2 using the FMS demonstrates competitive performance despite its simple process pipeline of inputting raw waveforms to a DNN and measuring cosine similarity using the output speaker embeddings.

5. Conclusion

In this paper, we have proposed various FMS-based methods to improve the existing RawNet system, which is neural speaker embedding extractor in which speaker embeddings are extracted directly from raw waveforms. The FMS uses a scale vector to perform scaling, where the dimension of the scale vector is identical to the number of filters. The FMS-based methods scale filters in feature maps to construct improved feature maps that focus on more important features in the frame-level feature map through addition, multiplication, or both. We applied various FMS-based methods to the output of each residual block. In addition, by replacing the first convolution layer with a sinc-conv layer, we achieved further improvements. The results of the evaluation performed using the original VoxCeleb1 protocol demonstrate an EER of 2.48 %, while the original RawNet reported EER of 4.80 %. In an evaluation using recently expanded evaluation protocols, the proposed method marginally outperformed the current state-of-the-art methods.

6. Acknowledgements

This research was supported by Projects for Research and Development of Police science and Technology under Center for Research and Development of Police science and Technology and Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (Grant No. PA-J000001-2017-101)

¹We report all performance values using the *cleaned* protocol.

7. References

- [1] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *Proc. Interspeech 2019*, pp. 1268–1272, 2019.
- [2] H. Muckenhirn, M. Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using cnns," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4884–4888.
- [3] 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)*. IEEE, 2018, pp. 5349–5353.
- [4] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," *arXiv preprint arXiv:1808.00158*, 2018.
- [5] E. Variiani, 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.
- [6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.
- [7] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.
- [8] Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," *Proc. Interspeech 2019*, pp. 4030–4034, 2019.
- [9] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.
- [10] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, 2020.
- [11] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3583–3587.
- [12] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Interspeech*, 2019.
- [13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [15] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.
- [16] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Interspeech*, 2018, pp. 3573–3577.
- [17] P. Safari and J. Hernando, "Self multi-head attention for speaker recognition," *Proc. Interspeech 2019*, pp. 4305–4309, 2019.
- [18] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "Cnn with phonetic attention for text-independent speaker verification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 718–725.
- [19] J. Zhang, N. Inoue, and K. Shinoda, "I-vector transformation using conditional generative adversarial networks for short utterance speaker verification," *Proceedings of INTERSPEECH, Hyderabad, India*, 2018.
- [20] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.
- [21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
- [23] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.
- [24] H.-S. Heo, J.-w. Jung, I.-H. Yang, S.-H. Yoon, H.-j. Shim, and H.-J. Yu, "End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification," *arXiv preprint arXiv:1902.02455*, 2019.
- [25] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018.
- [26] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," *arXiv preprint arXiv:1809.04437*, 2018.
- [27] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP*, 2019.
- [28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
- [29] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.
- [30] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [31] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [32] F. Chollet *et al.*, "Keras," <https://github.com/keras-team/keras>, 2015.
- [33] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.