INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Improving Time Delay Neural Network Based Speaker Recognition With Convolutional Block And Feature Aggregation Methods

Yu-Jia Zhang[1], *Yih-Wen Wang*[1], *Chia-Ping Chen*[1], *Chung-Li Lu*[2], *Bo-Cheng Chan*[2]

[1]National Sun Yat-Sen University, Taiwan
[2]Chunghwa Telecom Laboratories, Taiwan

m083040025@student.nsysu.edu.tw, m083040011@student.nsysu.edu.tw,
cpchen@cse.nsysu.edu.tw,
{chungli,cbc}@cht.com.tw

Abstract

In this paper, we develop a system that integrates multiple ideas and techniques inspired by the convolutional block and feature aggregation methods. We begin with the state-of-the-art speaker-embedding model for speaker recognition, namely the model of Emphasized Channel Attention, Propagation, and Aggregation in Time Delay Neural Network, and then gradually experiment with the proposed network modules, including bottleneck residual blocks, attention mechanisms, and feature aggregation methods. In our final model, we replace the Res2Block with SC-Block and we use a hierarchical architecture for feature aggregation. We evaluate the performance of our model on the VoxCeleb1 test set and the 2020 VoxCeleb Speaker Recognition Challenge (VoxSRC20) validation set. The relative improvement of the proposed models over ECAPA-TDNN is 22.8% on VoxCeleb1 and 18.2% on VoxSRC20.

Index Terms: speaker recognition, speaker verification, TDNN, ResNet, channel attention, feature aggregation.

1. Introduction

Integration of deep learning leads to breakthroughs in speaker recognition research. Specifically, X-vectors/Time Delay Neural Network (TDNN) [1] includes a deep neural network as front-end speaker embedding extractor and Probabilistic Linear Discriminant Analysis (PLDA) [2] or cosine similarity as back-end scorer. One of the active fields of research is to improve the TDNN. As TDNN is equivalent to 1-D Convolution Neural Network (CNN), researchers have also experimented with 2-D CNN, such as ResNet [3].

Recently, the state-of-the-art Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network (ECAPA-TDNN) [4] proposes to incorporate the elements of Res2Net [5] with TDNN. To model the interdependencies between the channels, ECAPA-TDNN applies the channel attention mechanisms of Squeeze-and-Excitation (SE) [6]. Moreover, the outputs of different dilation rates are concatenated to process expressive features.

In this work, we implement the ECAPA-TDNN as our baseline and propose further improvements to this basic architecture. Our experiments and analysis focus on bottleneck residual blocks, attention mechanisms, and feature aggregation methods. First, we adopted different bottleneck residual blocks to replace the Res2Block in ECAPA-TDNN, then substituted the SE with Efficient Channel Attention (ECA) [7] and Convolutional Block Attention Module (CBAM) [8], respectively. We also extend the feature aggregation method from multi-layer to hierarchical. Our final model, which includes SC-Block [9], SE, and hierarchical feature aggregation, achieves better performance than baseline on the Voxceleb1 [10] test set and VoxSRC20 [11] validation set.

This paper is organized as follows: Section 2 describes the different frame-level architectures. Section 3 introduces the experimental setup including training dataset, acoustic feature, training protocol, and evaluation protocol. Section 4 focuses on the results and analysis. Section 5 summarizes this work.

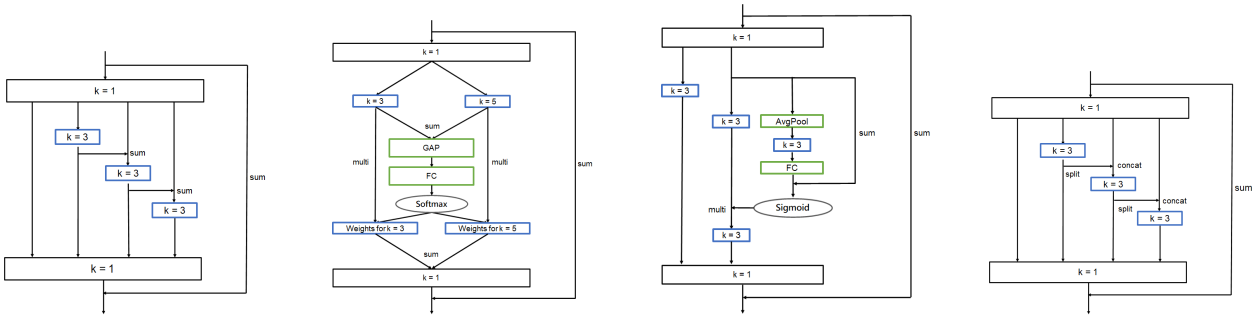
2. Network Architecture

In this section, we refer to ECAPA-TDNN [4, 12] to implement our baseline system and focus on the experiment of the frame-level architecture. The architecture of baseline is shown in Table 1. Conv1D means the 1-D convolution. BN stands for Batch Normalization and the ReLU is the non-linearity. FC(Emb.) denotes the fully connected layer where the speaker embedding is extracted. We fix the attentive statistics pooling [13] layer. Then we take the various residual blocks and attention mechanisms as a plug-and-play block to upgrade the baseline architecture. Also, we examine different feature aggregation methods. All the 2-D convolution in the residual blocks and attention mechanisms for the computer vision task is rescaled to TDNN layer

Table 1: *Baseline architecture. Layer denotes the network connection type. Structure describes the detail setting in layers. For example, "5,512" means 1-D convolution with a kernel size of 5 and 512 output channels, "*$fc, [64, 512]$*" indicates the output dimension of the two fully connected layers in an SE module, "*$D = 2$*" specifies the dilation rate of the TDNN layer in the second layer of Res2Block.*

#	Layer	Structure	Output shape	Input layer
0	Input	-	200×80	-
1	Conv1D + ReLU + BN	$5, 512$	200×512	0
2	SE-Res2Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ fc, [64,512] \end{bmatrix}, D = 2$	200×512	1
3	SE-Res2Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ fc, [64,512] \end{bmatrix}, D = 3$	200×512	2
4	SE-Res2Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ fc, [64,512] \end{bmatrix}, D = 4$	200×512	3
5	SE-Res2Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ fc, [64,512] \end{bmatrix}, D = 5$	200×512	4
6	Aggregation	-	200×2048	2,3,4,5
7	Conv1D + ReLU + BN	$1, 1536$	200×1536	6
8	Attentive Stat Pooling	-	3072	7
9	FC(Emb.)	3072×192	192	8
10	AAM-Softmax	$192 \times \#spks$	$\#spks$	9

http://dx.doi.org/10.21437/Interspeech.2021-356



(a) Res2Block (b) SK-Block (c) SC-Block (d) HS-Block

Figure 1: *The structures of experimental different bottleneck residual blocks. k : the kernel size of 1-D convolution. GAP : global average pooling.*

with different dilation rate or 1-D convolution for fewer parameters and larger receptive field to introduce these modules into the speaker recognition task.

2.1. Residual Blocks

In recent years, many studies focus on enlarging the receptive field of the convolutional layer on ResNet [3]. The first technique integrates the ResNet with the concept of inception [14], and proposes a split-transform-merge strategy called ResNext [15]. The cardinality is designed for processing different sizes of receptive fields to obtain multi-scale features. Moreover, Res2Net [5] further increases multi-scale feature extraction ability. The above concepts are similar to the TDNN, which gets a wide range of time information through convolution with different dilation rates. In ECAPA-TDNN, it merges the advantages of Res2Net and TDNN.

Therefore, we consider that different methods of obtaining multi-scale features have an impact on the final representation. In the following, we implement multiple approaches and search for the best one to replace Res2Net.

2.1.1. Res2Net

The bottleneck residual block used in ECAPA-TDNN is Res2Block which is shown in Figure 1(a). The characteristic is that the original input split into multiple groups according to channels. The output of one group is fed into the next group, and so on. All segments are concatenated as the final result.

2.1.2. SKNet

In selective kernel network (SKNet) [16], the bottleneck residual block is called SK-Block, as Figure 1(b). Unlike Res2Net, it does not split the input but uses different kernel sizes of convolution to conduct two transformations. The different outputs are fused via an element-wise summation. The soft attentions of different scales features achieve by GAP-FC-Softmax. These attentions estimate the importance of different outputs to get weighted multi-scale features.

2.1.3. SCNet

The self-calibration network (SCNet) [9] has the bottleneck residual block called SC-Block that integrates Res2Net and SKNet, shown in Figure 1(c). According to the channels, it split the input and use normal convolution and self-calibration. In self-calibration, the split input further converts into two different scale-spaces. One is the original space through convolution, the other obtains a smaller latent space as the spatial reference of the original space by average pooling (AvgPool), convolution, and FC. Finally, the weighted output passes to another convolution and is concatenated with the normal convolution output. Over the self-calibration calculation, it can exploit different portions of convolutional filters in a heterogeneous way, integrate the different scale spaces. Thus, it allows the network to learn the weighted multi-scale features and avoid certain unrelated information.

2.1.4. HSNet

The hierarchical-split network (HSNet) [17] is an extension of Res2Net and its bottleneck residual block is called HS-Block in Figure 1(d). Their difference is that the output would not be completely fed into the next group. Half of the output of each group is kept as the final result, the other half is concatenated with the input of the next group instead of summation. The concatenation prevents the feature representation from destroying. This approach improves the ability of multi-scale feature expression.

2.2. Attention Mechanisms

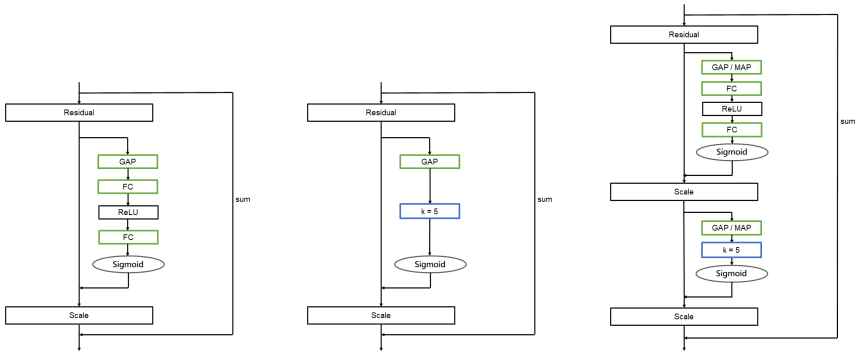
The network applies the attention mechanisms to focus on relative important information by calculating channel and spatial soft-attention. In many researches, it has been proven to effectively improve the effect of CNN. Hence, we believe that different attention mechanisms have an impact on the integration of features. Next, we try various attention mechanisms to enhance the ability to capture the important features.

2.2.1. SE

SE-Net [6] proposed an effective method to learn channel attention. It is shown in Figure 2(a). ECAPA-TDNN integrates the 1-D SE-block that rescales the frame-level features through squeeze and excitation. In the squeeze stage, the GAP averages frame-level features along the time domain. The excitation calculates channel-wise weights through two linear layers and non-linearity and then multiplies with the original features.

2.2.2. ECA

ECA-Net [7] is an improvement of SE-Net in Figure 2(b). It observes that dimension reduction operation in SE excitation causes part of the information loss and reduces the performance.



(a) SE (b) ECA (c) CBAM

Figure 2: *The structures of experimental different attention mechanisms.*

To address this problem, ECA replaces the two linear layers in excitation with a 1-D convolution layer. It guarantees efficiency and effectiveness by appropriately capturing local cross-channel interaction.

2.2.3. CBAM

CBAM [8] is an extension of SE-Net which additionally calculates spatial attention. It is shown in Figure 2(c). After the channel attention, the spatial attention inputs the channel-refined features and utilizes average pooling and max pooling along the channel axis to generate two different spatial features. Finally, the convolution is applied to form the spatial attention.

2.3. Feature Aggregation

[18, 19] demonstrate that integrating the different resolution features can extract the expressive speaker embeddings. Therefore, we implement the multi-layer approach in ECAPA-TDNN and expand it into the hierarchical one.

2.3.1. Multi-layer

ECAPA-TDNN adopts the multi-layer feature aggregation method to integrate the outputs of each SE-Res2Block. The principle is that SE-Res2Block outputs of different dilation rates are concatenated and pass the 1-D convolution to reduce the dimension.

2.3.2. Hierarchical

Inspired by DenseNet [20], we proposed the hierarchical technique as Figure 3. The outputs of all lower dilation rate SE-Res2Blocks are concatenated and then integrated by the 1-D convolution. Finally, they are fed into the higher dilation rate SE-Res2Blocks. The network can process information of different resolutions hierarchically to learn sophisticated representations and generate discriminative speaker embeddings.

3. Experimental setup

3.1. Datasets

All systems use the development of VoxCeleb2 [21] as a training dataset, which contains 5994 speakers. To make the system more robust, we randomly use 4 different data augmentation methods in every training step: select speech, music, noises from the MUSAN [22] dataset to add noise, and artificially reverberated via convolution with simulated RIRs [23].

3.2. Acoustic Feature

During training, we randomly extract 2 seconds of the chunk from the training files. Pre-emphasis is applied and then encodes the audio samples into 80-dimensional log Mel-filterbanks with a 25-ms frame window and a 10-ms frameshift. Mean and variance normalization (MVN) is performed by applying instance normalization to the network input. There is no Voice Activity Detector (VAD) applied.

3.3. Training protocol

Our implementation is adapted from Clova AI speaker recognition code [24] with PyTorch framework. All models use an initial learning rate of 0.001 and are reduced by 25% every 10 epoch with Adam optimizer. Except for the final model where we change the optimizer to AdamP [25] to stabilize the training. We use AAM-softmax loss [26] with a margin of 0.2 and softmax prescaling of 30, and train all models on GeForce GTX 1080ti for 100 epochs. The batch size is fixed at 256. A weight decay 2e-5 is applied. Convolutional frame layers are all set 512. The scale dimension s in the Res2Block and HS-Block is set to 8. The r for AvgPool in SC-Block is set to 8. The output size of speaker embedding is 192.

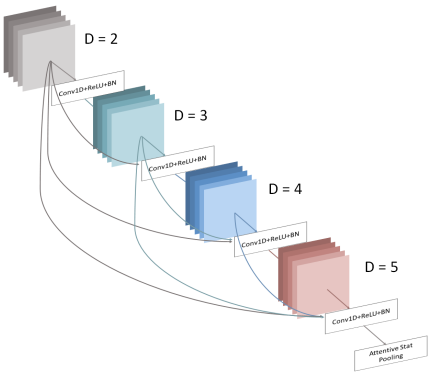


Figure 3: *Our hierarchical feature aggregation method.*

Table 2: *Our final architecture. We compared to the baseline architecture, replace the Res2Block with SC-Block. The hierarchical feature aggregation is also applied.*

#	Layer	Structure	Output shape	Input layer
0	Input	-	200×80	-
1	Conv1D + ReLU + BN	$5, 512$	200×512	0
2	SE-SC-Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ f_c,[64,512] \end{bmatrix}, D=2$	200×512	1
3	Aggregation	-	200×512	2
4	Conv1D + ReLU + BN	$1, 512$	200×512	3
5	SE-SC-Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ f_c,[64,512] \end{bmatrix}, D=3$	200×512	4
6	Aggregation	-	200×1024	2,5
7	Conv1D + ReLU + BN	$1, 512$	200×512	6
8	SE-SC-Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ f_c,[64,512] \end{bmatrix}, D=4$	200×512	7
9	Aggregation	-	200×1536	2,5,8
10	Conv1D + ReLU + BN	$1, 512$	200×512	9
11	SE-SC-Block	$\begin{bmatrix} 1,128 \\ 3,128 \\ 1,512 \\ f_c,[64,512] \end{bmatrix}, D=5$	200×512	10
12	Aggregation	-	200×2048	2,5,8,11
13	Conv1D + ReLU + BN	$1, 1536$	200×1536	12
14	Attentive Stat Pooling	-	3072	13
15	FC(Emb.)	3072×192	192	14
16	AAM-Softmax	$192 \times \#spks$	$\#spks$	15

3.4. Evaluation protocol

We evaluate our models on the Voxceleb1 [10] test set and VoxSRC20 [11] validation set, and report the performance metric: the Equal Error Rate (EER). The scoring of trials is using Euclidean distance between embeddings. We don't use any score calibration or normalization.

4. Result

Section 4.1 to 4.3 will conduct experiments on the various architectures and methods mentioned in Section 2. In Section 4.4, we combine improved methods into the final architecture. The first line of all result tables is the baseline setting.

4.1. Residual Blocks

We use different residual blocks to replace the original Res2Block in the baseline. The results are summarized in Table 3, which shows the improvement of SC-Block and HS-Block, especially SC-Block. It has relative improvements of 3.4% on VoxCeleb1 and 5.2% on VoxSRC20. SC-Block uses the calculation of self-calibration while enlarging the receptive field and gets the spatial attention of the context to avoid redundant information in the features. It can be considered as a frame and channel selective convolution method. Among the four methods, SK-Block has the worst effect. It uses different kernel size convolutions to achieve different transformations of the original input instead of split inputs by the channel. Part of the original representation is not retained and loses low-resolution features.

4.2. Attention Mechanisms

The baseline system uses SE to get channel interdependencies of contexts. Here we select two variants of attention mechanisms to replace SE. As shown in Table 4, the model with SE is still the best. We think that the 1-D convolution in ECA and CBAM conflicts with TDNN, resulting in redundant information.

Table 3: *EER of different residual blocks.*

Residual block	VoxCeleb1	VoxSRC20
Res2Block	1.49	4.46
SK-Block	1.70	4.64
SC-Block	**1.44**	**4.23**
HS-Block	1.45	4.26

Table 4: *EER of different attention mechanisms.*

Attention mechanism	VoxCeleb1	VoxSRC20
SE	**1.49**	**4.46**
ECA	1.74	4.65
CBAM	1.58	4.60

Table 5: *EER of different feature aggregation.*

Feature aggregation	VoxCeleb1	VoxSRC20
Multi-layer	1.49	4.46
Hierarchical	**1.41**	**4.18**

Table 6: *EER of our final architecture.*

Architecture	VoxCeleb1	VoxSRC20
Baseline	1.49	4.46
Ours($C = 512$)	1.28	4.04
Ours($C = 2048$)	**1.15**	**3.65**

4.3. Feature Aggregation

We use the hierarchical feature aggregation compare to the multi-layer. Table 5 presents that the hierarchical method has relative improvements of 5.4 % on VoxCeleb1 and 6.3 % on VoxSRC20. It further confirms the idea of [18, 19] that concatenating and extracting multiple temporal scale features can enhance discriminability of speaker embeddings.

4.4. Final Architecture

Based on the former experiments, we design the final architecture which consists the residual block: SC-Block, attention mechanism: SE, and feature aggregation: hierarchical. We use two different convolution channel size settings are $C = 512$ and $C = 2048$, respectively. The final architecture is shown in Table 2 and the results are summarized in Table 6. We can see that our architecture can significantly boost the performance compared to the baseline. With $C = 512$, it has 14.1% relative improvements on VoxCeleb1, and 9.4% on VoxSRC20. With $C = 2048$, it relatively improves 22.8 % on VoxCeleb1, and 18.2% on VoxSRC20. The results also demonstrate that simply increasing the convolution channel size can improve the performance on VoxSRC20.

5. Conclusion

In this study, we adapt the frame-level layer architecture contain residual blocks, attention mechanisms, and feature aggregation based on the state-of-the-art ECAPA-TDNN model of speaker recognition. We replace the Res2Block with SC-Block and propose the hierarchical feature aggregation method to build our final model. It relatively improves 22.8% on VoxCeleb1 test set and 18.2% on VoxSRC20 validation set compared to our baseline.

6. References

[1] 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.

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

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.90

[4] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2650

[5] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A New Multi-Scale Backbone Architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, p. 652–662, Feb 2021. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2938758

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00745

[7] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.01155

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," *Lecture Notes in Computer Science*, p. 3–19, 2018. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-01234-2_1

[9] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 096–10 105.

[10] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," *Interspeech 2017*, Aug 2017. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[11] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "VoxSRC 2020: The Second VoxCeleb Speaker Recognition Challenge," 2020.

[12] J. Thienpondt, B. Desplanques, and K. Demuynck, "The IDLAB VoxCeleb Speaker Recognition Challenge 2020 System Description," 2020.

[13] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Interspeech*, 2018, pp. 2252–2256.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," 2016.

[15] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017.

[16] X. Li, W. Wang, X. Hu, and J. Yang, "Selective Kernel Networks," 2019.

[17] P. Yuan, S. Lin, C. Cui, Y. Du, R. Guo, D. He, E. Ding, and S. Han, "HS-ResNet: Hierarchical-Split Block on Convolutional Neural Network," 2020.

[18] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System," in *Proc. Interspeech 2019*, 2019, pp. 361–365. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1489

[19] Y. Jung, S. M. Kye, Y. Choi, M. Jung, and H. Kim, "Improving Multi-Scale Aggregation Using Feature Pyramid Module for Robust Speaker Verification of Variable-Duration Utterances," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1025

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.243

[21] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," *Interspeech 2018*, Sep 2018. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1929

[22] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 5220–5224.

[24] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[25] B. Heo, S. Chun, S. J. Oh, D. Han, S. Yun, G. Kim, Y. Uh, and J.-W. Ha, "AdamP: Slowing Down the Slowdown for Momentum Optimizers on Scale-invariant Weights," 2020.

[26] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.