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

Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition

Magdalena Rybicka[1*], *Jesús Villalba*[2,3], *Piotr Żelasko*[2,3], *Najim Dehak*[2,3], *Konrad Kowalczyk*[1]

[1]AGH University of Science and Technology, Institute of Electronics, Kraków, Poland
[2]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA
[3]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, USA
{mrybicka, konrad.kowalczyk}@agh.edu.pl, {jvillal7, pzelasko, ndehak3}@jhu.edu

Abstract

Modeling speaker embeddings using deep neural networks is currently state-of-the-art in speaker recognition. Recently, ResNet-based structures have gained a broader interest, slowly becoming the baseline along with the deep-rooted Time Delay Neural Network based models. However, the scale-decreased design of the ResNet models may not preserve all of the speaker information. In this paper, we investigate the SpineNet structure with scale-permuted design to tackle this problem, in which feature size either increases or decreases depending on the processing stage in the network. Apart from the presented adjustments of the SpineNet model for the speaker recognition task, we also incorporate popular modules dedicated to the residual-like structures, namely the Res2Net and Squeeze-and-Excitation blocks, and modify them to work effectively in the presented neural network architectures. The final proposed model, i.e., the SpineNet architecture with Res2Net and Time-Squeeze-and-Excitation blocks, achieves remarkable Equal Error Rates of 0.99 and 0.92 for the Extended and Original trial lists of the well-known VoxCeleb1 dataset.

Index Terms: deep neural networks, SpineNet model, scale-permuted network, ResNet model, speaker recognition

1. Introduction

Current state-of-the-art in speaker recognition is to model speaker characteristics using deep neural networks (DNN), from which embeddings – commonly referred to as x-vectors [1] – are extracted. This approach has been shown in numerous studies [1, 2, 3] to outperform the well-established i-vector model [4]. Baseline DNN architectures include the so-called Time Delay Neural Networks (TDNN) [1], as well as their two modifications known as the Extended TDNN [5] and Factorized TDNN [6]. Recently, we have observed a rapid increase of popularity of the ResNet-based structures, with model adjustments presented e.g. in [2, 7, 8], which often offer an improved performance over the aforementioned baseline models. In [9] the authors point out that the scale-decreased design of the ResNet model may cause a removal of useful information. In order to overcome this problem, they propose a scale-permuted network design [9], where feature resolution and dimension can change arbitrarily as it is processed through the network. It outperformed i.a. ResNet with Feature Pyramid Network [10].

In this paper, we adjust the scale-permuted SpineNet structure [9] and incorporate it in the DNN-based speaker recognition model. We show that the multi-scale feature representation of SpineNet, in which input to the pooling layer is merged from several prior layers with various feature resolution, overcomes some of the limitations encountered by the scale-decreased models such as ResNet. In the context of speaker recognition, multi-scale feature resolution has been studied in [11, 12, 13, 14] for utterances of variable length, achieving notable improvement. In addition, we modify two existing residual-like structures such as Res2Net block [15] and Squeeze-and-Excitation (SE) block [16], and show that the presented modules further improve system performance. The final proposed SpineNet model with Res2Net and Time-Squeeze-and-Excitation blocks achieves highly competitive results for the trial lists of the VoxCeleb1 dataset, outperforming the known models such as [2].

[*]work performed while at Center for Language and Speech Processing, Johns Hopkins University.

2. Deep Neural Network Structures

2.1. ResNet architectures

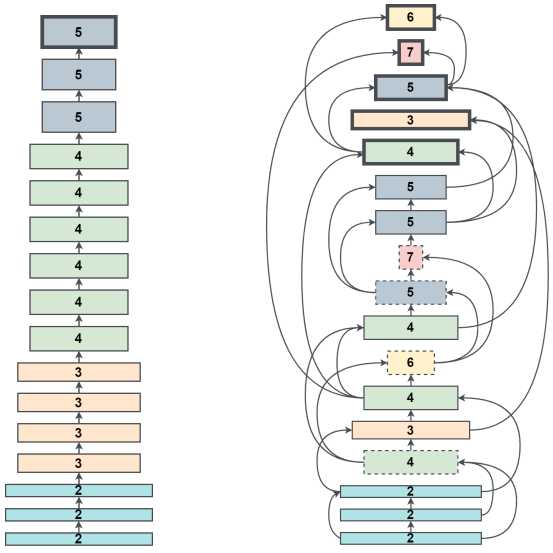
In this work, we consider three models based on the well-known ResNet architecture [17] for speaker embedding extraction. ResNet-34 and ResNet-50 [18] follow the so-called scale-decreased design, in which the size of the feature map is reduced as it is processed by the network. In both models, the sequence of the main building blocks is similar. The input feature map is passed through a 64-channel 3×3 convolutional layer with stride=1. Next, the output features are processed by the so-called residual part of the structure, which is presented in Figure 1a for ResNet-50. The residual part of the ResNet-34 exhibits the same block sequence, with the difference that the bottleneck residual blocks are replaced with the basic residual blocks [18]. As shown in Figure 1a, residual blocks are distributed across several levels (2-5). At the beginning of each level–starting at level 3– the feature maps are downsampled by 2. Thus, a block at level l has feature maps downsampled by a factor 2^{l-2}. Table 1 presents the resulting feature map size and the number of base channels for the blocks at each level. Note that for the basic residual block, the number of output channels is equal to the number of the base channels, while the bottleneck block increases the output channels by 4. In both networks, the output of the residual part is passed to the statistics pooling, followed by the fully connected layer along with the softmax layer. In our experiments, we also considered a light version of the ResNet-34 model, in which the number of channels of all blocks were decreased by 4. Hereafter, we will refer to this structure as Thin-ResNet-34.

2.2. SpineNet architectures

SpineNet structure belongs to the family of scale-permuted meta-architectures [9]. The inner connections, block order and

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-1163



(a) *ResNet-50* (b) *SpineNet-49*

Figure 1: *The structure of the residual part of (a) ResNet-50 and (b) SpineNet-49 networks. In diagrams, bottleneck blocks are marked with solid lines, basic residual blocks are marked with dotted lines. Blocks with bold line contours (located in the top of the architectures) represent the output blocks, while the number inside each block indicates its level.*

Table 1: *Sizes of features for blocks corresponding to the levels* $l = 2, 3, 4, 5, 6,$ *and* 7 *for the ResNet and SpineNet models. The feature map size is given in terms of F (feature) and T (time) lengths, while the feature dimension is given by the number of base channels.*

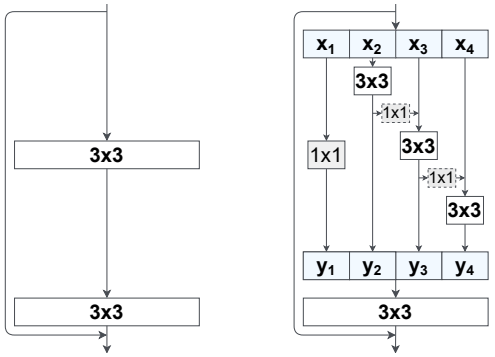
Block level	Feature map size	No. base channels	
		ResNet	SpineNet
2	F × T	64	64
3	F/2 × T/2	128	128
4	F/4 × T/4	256	256
5	F/8 × T/8	512	256
6	F/16 × T/16	—	256
7	F/32 × T/32	—	256

their type is derived by Neural Architecture Search in [9], with ResNet-50 incorporated as a baseline.

Similar to the ResNet model, the network input is first processed by the 3×3 convolutional layer with stride of 1×1. The next part of the structure is presented in Figure 1b. It consists of stem scale-decreased and learned scale-permuted segments.

The stem network part is represented by the first two bottleneck blocks at the second level, whose outputs are used as candidate input features for the scale-permuted segment. In the scale-decreased network, block sequence follows a fixed order, where block level is kept unchanged or it is increased with the network processing flow.

The scale-permuted part is build of blocks that arbitrarily increase or decrease its level with the network processing flow. In this segment, blocks accept 2 input connections, where output blocks, indicated by bold contour lines in Figure 1b, accept up to 3 inputs. Input features are fused by an element-wise addition. Since the connections between the blocks are cross-scale, each connection consists of 3 components: (i) 1×1 convolution, which reduces the number of channels by a factor $\alpha=0.5$ compared with the number of base channels of the block from which the connection is made, (ii) feature map resampling operation;



(a) *Basic residual block* (b) *Basic Res2Net block*

Figure 2: *Basic residual block and its proposed Res2Net adaptation. Example presented for the scale* $s = 4$.

and (iii) another 1×1 convolution which transforms the channel number to the target size. The target number of channels for the basic residual blocks is equal to the number of base channels, while the target number of channels for the bottleneck blocks is 4 times larger. In contrast to the ResNet structures, SpineNet incorporates both types of residual blocks. Table 1 presents the feature map sizes and the number of base channels associated with each level of the structure. In the resampling part, the upsampling operation is performed by the nearest-neighbor interpolation. The downsampling is achieved by the convolution of size 3×3, with stride 2. If necessary the convolution is followed by maximum pooling with a 3×3 kernel and a stride of 2 or alternatively with a 5×5 kernel and a stride of 4.

Output block features (marked with bold contour lines in the Figure 1b) are processed by 1×1 endpoint convolutions to obtain the common number of channels, which we set to 256. Next, the feature maps are upsampled with nearest-neighbour interpolation to match the feature map size at the lowest level. Representations from the different levels are merged with a point-wise average operation and are forwarded to the statistics pooling followed by the fully connected and softmax layers.

In this work, we also present the results for the two modifications of the SpineNet-49 structure, namely the so-called Thin-SpineNet-49 and SpineNet-49S. The former follows similar modification as in ResNet, i.e. the number of channels of all layers is reduced 4 times (scaling factor of 0.25), including the number of the endpoint filters. The latter model, SpineNet-49S, represents the intermediate structure between the Thin-SpineNet-49 and SpineNet-49, where filter dimensions are decreased using a scaling factor of 0.66 (except for the input convolution) and the number of endpoint channels is set to 128. In the original paper [9], the SpineNet-49S has an associated factor of 0.65, which we modify in order to obtain the desired number of channels such that the modifications described in the next two subsections were feasible.

2.3. Res2Net module

In this work, we incorporate the so-called Res2Net modules [15] into ResNet and SpineNet. Res2Net introduces a new dimension - scale s, which increases the receptive field and granular level of the bottleneck residual block.

Res2Net blocks were proposed as substitute for the residual bottleneck blocks in structures like ResNet-50 and larger. Since SpineNet also includes basic residual blocks, we adapted the original Res2Net structure to the basic block. Our modification is presented in Figure 2. Input of the basic Res2Net block is

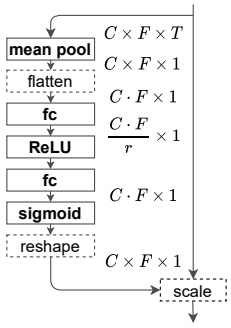


Figure 3: *Diagram of the Time-Squeeze-and-Excitation (T-SE) module, where C, F, and T denote the number of input channels, the frequency dimension, and the time dimension, while r is the reduction ratio.*

split evenly into s parts along the channel dimension. Then, each x_i group, where $i = 1, ..., s$, is processed by a separate convolutional layer with w output channels. The corresponding output y_i can be expressed as follows:

$$y_i = \begin{cases} \mathrm{K}_{1\times1}(x_i) & \text{if } i = 1 \\ \mathrm{K}_{3\times3}(x_i) & \text{if } i = 2 \\ \mathrm{K}_{3\times3}(x_i + y_{i-1}) & \text{if } s \geq i > 2, w = C_{\mathrm{in}}/s \\ \mathrm{K}_{3\times3}(x_i + \mathrm{K}_{1\times1}(y_{i-1})) & \text{if } s \geq i > 2, w \neq C_{\mathrm{in}}/s \end{cases} \quad (1)$$

where $\mathrm{K}_{1\times1}$ and $\mathrm{K}_{3\times3}$ are convolutional layers with a 1×1 and 3×3 kernels respectively; and C_{in} are the block input channels. The inner convolution channels w can be set equal to the number of input channels, or alternatively it can be increased to preserve/extend network complexity. For the latter case, we need projection convolutions $\mathrm{K}_{1\times1}$ in the inner residual connections to match the channel dimensions and thereby enable the addition operation. Concatenated y_i features are passed to another convolutional layer (kernel of 3×3). In this work, we incorporate the scale $s = 4$ and we set $w = 26$ for the inner convolutions for residual blocks with 64 base channels (level 2). This increased the total number of inner channels from 64 to 104 w.r.t. the standard residual blocks. We observed that such an increase in the Res2Net was required to maintain the performance. The value of w for the blocks with a higher number of base channels is increased proportionally to the block size. We will refer to the SpineNet with Res2Net blocks as Spine2Net.

2.4. (Time-)Squeeze-and-Excitation blocks

Squeeze-and-Excitation (SE) blocks [16] constitute a common approach to re-calibrate channel dependencies. In case of speaker recognition, the frequency dependencies are also of high importance. Therefore, we enhance the modelling capabilities of SE blocks by introducing the Time-Squeeze-and-Excitation (T-SE) module [19]. The T-SE model is similar to the SE model, however, the average pooling is applied only along the time dimension, instead of the global pooling of the entire feature map. The algorithm pipeline is presented in Figure 3. The squeeze operation produces the channel- and frequency-wise descriptor by applying mean pooling along the time axis. This process is followed by an excitation step, in which calibration weights are estimated. These weights are computed by a dimensionality reduction layer with reduction factor r, ReLU non-linearity, and fully-connected layer with sigmoid activation. Obtained scale values are then used to re-calibrate the feature maps. Note that, original SE pools over time and frequency axis producing only a single scale value per channel. Thus, all frequency bins are scaled by the same value, while T-SE produces a different scaling per bin.

3. Experimental Evaluation and Results

3.1. Datasets, system framework and evaluation measures

This section presents the datasets and general framework of the evaluated speaker recognition systems. Neural networks were trained on VoxCeleb2 [20] with 6112 speakers. Utterances derived from the same video were concatenated and extended with 3 types of noise augmentations: music, environmental noise, babble speech from the MUSAN corpus [21], and reverberation with three sets of room impulse responses (RIRs) [22]. The test set is based on the VoxCeleb1 corpus [23] and clean versions of trials: Extended (VoxCeleb1-E), Hard (VoxCeleb1-H), and Original (VoxCeleb1-O) [24]. One epoch of the training consisted of randomly selected, augmented 4 s chunks in the number equal to the number of all augmented training utterances. Each network was trained for 70 epochs.

Input features were 80-dimensional log-Mel filter-banks extracted from 25 ms sliding window with 10 ms shift, and mean-normalization over a 3 s window. Speech frames were selected with Kaldi energy-based Voice Activity Detector [25]. The neural network was trained with Additive Angular Softmax [26] loss with scale $s_{\mathrm{AAS}} = 30$ and margin $m_{\mathrm{AAS}} = 0.3$. The margin was linearly increased from 0 to 0.3 during the first 20 epochs. Speaker embeddings were extracted from the penultimate fully connected layer, which yields a feature vector of length 256. In the system backend, we used cosine scoring as it provided better results than PLDA. The network structure was implemented in PyTorch [27] and it was trained with Adam optimizer [28] along with an exponential learning rate scheduler with an initial value of 0.05 [29, 30]. For experiments incorporating the SE blocks, the reduction factor was set to $r_{\mathrm{SE}} = 16$. T-SE blocks had a larger value of $r_{\mathrm{T\text{-}SE}} = 256$, as the T-SE blocks significantly enlarge the network size.

As evaluation measures, we used the Equal Error Rate (EER), reported in %, and minimum Detection Cost Function [31] with $P_{tar} = 0.05$ (DCF5) and $P_{tar} = 0.01$ (DCF1). The FLOP values were calculated for a 3 s utterance, with multiply-add counted as a single operation.

3.2. Results and discussion

Table 2 presents the results of a preliminary comparison of Thin versions of ResNet-34 and SpineNet-49. First, we compare two structures based only on single-scale features (lines 1-2). The output from the last layer in Thin-ResNet-34 and the output from the last block at the 5th level from the Thin-SpineNet-49 (denoted as Thin-SpineNet-49-5) were taken as single feature representations for pooling input. For fair comparison, Thin-SpineNet-49-5 representations were directly forwarded to the next layers without endpoint post-processing. Both structures provided similar performance, with a slight predominance of the ResNet model. The second experiment (lines 3-4) presents the gain offered by using multi-scale features. Thin-ResNet-34 modified to produce multi-scale features is denoted as Thin-ResNet-34-345. To this end, we incorporated the features from the last layer of blocks at the 3, 4, and 5th levels. As for the SpineNet models, the features were processed with a 1×1 convolutional layer transforming the channel number to 64 and upsampling the feature maps to the size of the 3rd level, followed by an average across levels. We observe that, for ResNet, multi-scale feature fusion did not significantly improve, except for

Table 2: *Results of experimental evaluation of Thin SpineNet and ResNet structures on the VoxCeleb1 test datasets.*

			VoxCeleb1-E			VoxCeleb1-H			VoxCeleb1-O		
Network	# Params	# FLOPs	EER	DCF5	DCF1	EER	DCF5	DCF1	EER	DCF5	DCF1
Thin-ResNet-34	3.6M	1.7G	1.90	0.119	0.200	3.25	0.189	0.298	2.05	0.149	0.240
Thin-SpineNet-49-5	3.9M	1.4G	1.95	0.123	0.209	3.28	0.189	0.298	2.07	0.135	0.211
Thin-ResNet-34-345	4.2M	1.7G	1.89	0.120	0.208	3.27	0.191	0.299	1.99	0.135	0.217
Thin-SpineNet-49	4.3M	1.7G	1.83	0.117	0.196	3.20	0.184	0.293	1.84	0.127	0.209

Table 3: *Results of experimental evaluation of SpineNet and ResNet structures, along with introduced modifications including Res2Net modules, Squeeze-and-Excitation (SE) blocks, and the Time-Squeeze-and-Excitation (T-SE) blocks on the VoxCeleb1 test datasets.*

			VoxCeleb1-E			VoxCeleb1-H			VoxCeleb1-O		
Network	# Params	# FLOPs	EER	DCF5	DCF1	EER	DCF5	DCF1	EER	DCF5	DCF1
ResNet-34	25.5M	27.3G	1.19	0.078	0.140	2.27	0.137	0.219	1.35	0.088	0.146
ResNet-50	35.6M	30.7G	1.30	0.082	0.150	2.33	0.142	0.235	1.44	0.100	0.173
SpineNet-49S	13.5M	11.2G	1.25	0.079	0.138	2.29	0.137	0.226	1.11	0.069	0.120
SpineNet-49	28.6M	26.0G	1.17	0.074	0.129	2.14	0.129	0.213	1.11	0.088	0.125
Res2Net-34	26.1M	27.6G	1.16	0.074	0.130	2.17	0.128	0.218	1.18	0.078	0.115
Res2Net-50	35.7M	32.0G	1.09	0.068	0.122	2.00	0.119	0.195	1.14	0.081	0.116
Spine2Net-49S	13.5M	11.3G	1.13	0.073	0.131	2.18	0.130	0.210	1.02	0.077	0.137
Spine2Net-49	28.8M	26.2G	1.10	0.071	0.127	2.18	0.132	0.216	1.09	0.070	0.116
SE-Res2Net-50	38.2M	32.0G	1.24	0.080	0.140	2.45	0.141	0.225	1.31	0.084	0.132
SE-Spine2Net-49S	14.0M	11.3G	1.09	0.069	0.122	2.11	0.124	0.205	1.05	0.066	0.104
SE-Spine2Net-49	29.8M	26.2G	1.04	0.067	0.119	2.07	0.121	0.206	1.14	0.066	0.098
T-SE-Res2Net-50	88.1M	32.1G	1.05	0.067	0.117	1.95	0.113	0.196	1.12	0.071	0.103
T-SE-Spine2Net-49S	26.0M	11.3G	1.08	0.070	0.124	2.09	0.124	0.204	1.08	0.074	0.127
T-SE-Spine2Net-49	58.0M	26.2G	0.99	0.065	0.112	1.95	0.117	0.192	0.92	0.068	0.105

VoxCeleb1-O. On the other hand, the Thin-SpineNet-49 benefited from the multi-scale representations, outperforming the other structures in all three test datasets.

Table 3 presents the results of a set of experiments of architectures with large, more complex structures and channel numbers as in their original form. We report on the results of four experiments: the baseline results (original structures), additional incorporation of the Res2Net modules, incorporation of the Squeeze-and-Excitation (SE) blocks on top of the previous alterations, and incorporation of the presented Time-Squeeze-and-Excitation (T-SE) blocks instead of SE blocks.

The first block of the table compares four basic structures, namely ResNet-34, ResNet-50, SpineNet-49S, and SpineNet-49. Comparing ResNets, ResNet-50 did not provide any gain over ResNet-34, despite having more learnable parameters. SpineNet-49S improved over ResNet-50 and presented competitive results to the ResNet-34 model. Note that SpineNet-49S has half of parameters and FLOPs than ResNet-34. SpineNet-49 clearly outperformed both ResNet structures for all the test datasets. Furthermore, the SpineNet-49 structure provided better performance with a lower number of FLOPs than ResNet-34, although it has more parameters. It is important to note that the reported number of parameters and FLOPs does not imply directly an improved training speed of the network but rather the structure effectiveness with respect to the model size.

The second block of the table introduces the Res2Net blocks to the described structures. In all four models, this modification provided a notable gain for speaker recognition measures–except the Hard condition for the Spine2Net-49, which achieved a comparable result than its original model. The Res2Net-34 model presents the effectiveness of the proposed Res2Net module adaptation for the basic residual block, reporting a clear gain over ResNet-34. Res2Net-50 achieved significantly better results than the ResNet-50 with 21% and 33% of relative improvement for EER and DCF1 for the VoxCeleb1-O dataset. Similar as in the previous evaluation, SpineNet-49S appears to be a competitive structure, outperforming the Res2Net-34 in nearly all test scenarios, while having less than half of the computational cost. Since Res2Net-50 presents clearly better results than Res2Net-34, in the following experiments we focus on the Res2Net-50 model only. In the third block of the table, we incorporate SE modules to the previous structures. The reported results show evidently accuracy degradation for the SE-Res2Net-50, whereas both Spine2Net structures clearly benefit from introduction of the re-calibration module. In final evaluation, the SE modules are replaced with the T-SE modules. The gain achieved by this replacement is evident. The least relative improvement can be observed for the T-SE-Spine2Net-49S architecture, nevertheless, the model still performs competitive, offering high recognition accuracy and low complexity. Note that the T-SE block strongly increases the number of network parameters, whereas the number of FLOPs is kept almost intact. Among all compared models, the last proposed structure, namely the T-SE-Spine2Net-49 model, provides an outstanding performance, outperforming all other systems.

4. Conclusions

This paper investigated the application of the scale-permuted architecture known as SpineNet for the speaker recognition task. Having adjusting the model, we incorporated Res2Net and Squeeze-and-Exciation modules, and proposed their modifications to achieve superb performance in the studied task. The results of experiments demonstrate that speaker recognition accuracy benefits from adopting the SpineNet structure and its multi-scale feature representation. Furthermore, the proposed Res2Net and T-SE modules further boost its performance.

5. Acknowledgements

This research was supported by the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union, and in part by PLGrid Infrastructure.

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)*, 2018, pp. 5329–5333.

[2] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. A. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.

[3] M. K. Nandwana, J. van Hout, M. McLaren, A. Stauffer, C. Richey, A. Lawson, and M. Graciarena, "Robust Speaker Recognition from Distant Speech under Real Reverberant Environments Using Speaker Embeddings," in *Proc. Interspeech 2018*, 2018, pp. 1106–1110.

[4] 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, 2010.

[5] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-speaker Conversations Using X-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[6] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747.

[7] M. Rybicka and K. Kowalczyk, "On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition," in *Proc. Interspeech 2020*, 2020, pp. 3805–3809.

[8] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[9] X. Du, T. Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X. Song, "SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 589–11 598.

[10] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[11] 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," in *Proc. Interspeech 2020*, 2020, pp. 1501–1505.

[12] 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.

[13] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut Connections Based Deep Speaker Embeddings for End-to-End Speaker Verification System," in *Proc. Interspeech 2019*, 2019, pp. 2928–2932.

[14] A. Hajavi and A. Etemad, "A Deep Neural Network for Short-Segment Speaker Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2878–2882.

[15] 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.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[17] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[18] 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)*, pp. 770–778, Jun 2016.

[19] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. Garcia-Perera, and N. Dehak, "Feature Enhancement with Deep Feature Losses for Speaker Verification," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7584–7588.

[20] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[21] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[22] *Room Impulse Response and Noise Database*, accessed February 17, 2021. [Online]. Available: http://www.openslr.org/28/

[23] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[24] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[26] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[29] J. A. Villalba Lopez, D. Garcia-Romero, N. Chen, G. Sell, J. Borgstrom, A. McCree, L. P. Garcia Perera, S. Kataria, P. S. Nidadavolu, P. Torres-Carrasquiilo, and N. Dehak, "Advances in Speaker Recognition for Telephone and Audio-Visual Data: the JHU-MIT Submission for NIST SRE19," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 273–280.

[30] J. Villalba and N. Dehak, *The JHU System Description for SDSV2020 Challenge*, accessed June 10, 2021. [Online]. Available: https://sdsvc.github.io/2020/descriptions/Team10_Both.pdf

[31] NIST, *NIST 2018 Speaker Recognition Evaluation Plan*, accessed June 10, 2021. [Online]. Available: https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf