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

Adaptive Convolutional Neural Network for Text-Independent Speaker Recognition

Seong-Hu Kim, Yong-Hwa Park

School of Mechanical Engineering, Korea Advanced Institute of Science and Technology
seonghu.kim@kaist.ac.kr, yhpark@kaist.ac.kr

Abstract

In text-independent speaker recognition, each speech is composed of different phonemes depending on spoken text. The conventional neural networks for speaker recognition are static models, so they do not reflect this phoneme-varying characteristic well. To tackle this limitation, we propose an adaptive convolutional neural network (ACNN) for text-independent speaker recognition. The utterance is divided along the time axis into short segments with small fluctuating phonemes. Frame-level features are extracted by applying input-dependent kernels adaptive to each segment. By applying time average pooling and linear layers, utterance-level embeddings extraction and speaker recognition are performed. Adaptive VGG-M using 0.356 seconds segmentation shows better speaker recognition performance than baseline models, with a Top-1 of 86.51% and an EER of 5.68%. It extracts more accurate frame-level embeddings for vowel and nasal phonemes compared to the conventional method without overfitting and large parameters. This framework for text-independent speaker recognition effectively utilizes phonemes and text-varying characteristic of speech.

Index Terms: speaker recognition, text-independent, adaptive convolutional neural network, frame-level speaker embedding

1. Introduction

Text-independent speaker recognition technology has been developed using deep neural networks (DNNs) [1, 2, 3, 4, 5, 6]. There have been many researches to reveal what factors affect speaker embedding networks' performances. Some of these researches have shown that the performance varies with phonemes [7, 8, 9]. Especially, vowels and nasals, generated from speakers' own vocal structure, have a major influence. However, static neural networks, which are mainly used in previous text-independent speaker recognition methods, extract speaker information without considering this characteristic of phonemes depending on the random text. We hypothesize that if the model works adaptively along the time axis, it could capture the phoneme-varying characteristics of speech. Thus, more accurate frame-level embeddings can be extracted and speaker recognition performance will be enhanced accordingly.

Content-adaptive neural networks have been mainly studied in the computer vision tasks and recently also applied to language and speech tasks. It can be broadly categorized into two types: self-attention module and adaptive convolutional neural network (ACNN). Self-attention module calculates an attention map of the output feature of previous layer [10, 11]. The feature multiplied by the attention map is used as the input of the next layer. In speaker recognition, several studies suggest applying it to the spectrogram or the pooling layer [12, 13, 14, 15]. ACNN, another type of content-adaptive neural network, is a module in which kernels of a neural network change according to its input or contents [16, 17, 18, 19]. It has been mainly conducted for the computer vision [20, 21, 22] and NLP [23, 24] applications, and attempts to apply it in speaker recognition are insufficient so far [25]. Both content-adaptive neural networks show positive results in various fields, especially vision area. However, unlike images that have a fixed size, the speech feature is in the time-frequency domain with variant time length, so a new methodology applicable to speaker recognition different from the computer vision is required.

In this paper, we propose a new adaptive convolutional neural network (ACNN) for text-independent speaker recognition in which CNN kernels change according to speech feature segments. The main contributions of this work are as follows:

1. We propose an architecture and framework that apply the ACNN module by segmenting the speech along the time axis to consider the phonemes-varying characteristic.
2. The ACNN module uses time and frequency scaling maps suitable for audio data such as speech, not image.
3. We propose the first analysis how the ACNN works according to phonemes in text-independent speaker recognition compared to the static model.

The remainder of the paper is organized as follows. Section 2 introduces an adaptive convolutional neural network for speaker recognition. Section 3 describe experiment setup and details. Section 4 shows the experiment results and discussion. Finally, Section 5 presents conclusions.

2. Adaptive Convolutional Neural Network Architecture

In this section, we proposed the model using a dividing layer which divides the input with a specific time length. Each segment is applied to the speaker recognition model using the ACNN module. Results in the same utterance are pooled along the time axis.

2.1. Adaptive Convolutional Neural Network Module

Each axis of time-frequency data such as spectrogram represents different physical dimensions, so we use two scaling maps, which are frequency and time domain, to each axis for the adaptive kernel in the ACNN module. The structure of proposed ACNN module for speaker recognition is shown in Figure 1.

The ACNN module starts by generating the two scaling matrices from the input $X \in \mathbb{R}^{C_{in} \times H \times W}$ with pooling layers, where C_{in} is the number of input channels, C_{out} is the number of output channels, H and W are the input shape, respectively. The frequency-domain scaling matrix $M_f(X) \in \mathbb{R}^{C_{in} \times K_H}$ and the time-domain scaling matrix $M_t(X) \in \mathbb{R}^{C_{in} \times K_W}$ are

Copyright © 2021 ISCA http://dx.doi.org/10.21437/Interspeech.2021-65

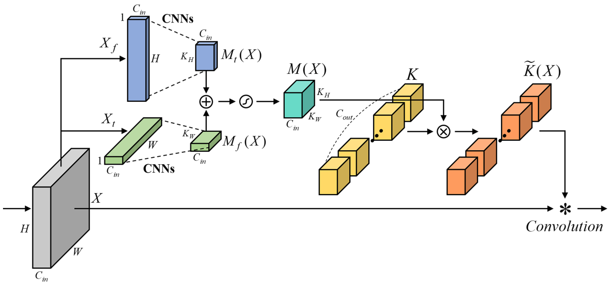


Figure 1: *Structure of the adaptive convolutional neural network module for speaker recognition. The adaptive kernel $\tilde{K}(X)$ is created by element-wise multiplication of each output channel C_{out} of the content-invariant kernel K with $M(X)$, and then it is used for convolution on X.*

generated as follows.

$$M_f(X) = conv_{(K_H,H,3)}(ReLU(conv_{(H,H,3)}(X_f{}^T))) \quad (1)$$

$$M_t(X) = conv_{(K_W,W,3)}(ReLU(conv_{(W,W,3)}(X_t{}^T))) \quad (2)$$

where *conv* is 1D-CNN, its subscript is the weight size as (output channel, input channel, kernel), K_H and K_W are the kernel shape. $M_f(X)$ is computed from a matrix $X_f \in \mathbb{R}^{C_{in}\times H}$ generated by global time-average pooling which computes the mean along the time axis. $M_t(X)$ is computed from a matrix $X_t \in \mathbb{R}^{C_{in}\times W}$ generated by global frequency-average pooling which computes the mean along the frequency axis. All convolutions have kernel size 3 with 1 stride and 1 zero-padding, and *ReLU* denotes the nonlinear function ReLU. To generate 3D scaling matrix, the two scaling matrices are broadcasted to $\mathbb{R}^{C_{in}\times K_H \times K_W}$ to match the shapes, and the element-wise summation for efficient gradient flow [10, 26] is applied for combining them. The sigmoid function σ is applied to the combined scaling matrix to determine the final scaling matrix $M(X) \in \mathbb{R}^{C_{in}\times K_H\times K_W}$ having a value between 0 and 1 as follows.

$$M(X) = \sigma(M_f(X) \oplus M_t(X)) \quad (3)$$

where $\oplus$ denotes element-wise summation after broadcasting 2D scaling matrixes to 3D. The generated 3D scaling matrix is multiplied element-wise by each output channel of the 4D content-invariant kernel $K \in \mathbb{R}^{C_{out}\times C_{in}\times K_H\times K_W}$ which is trainable parameter. It is the adapted kernel $\tilde{K}(X) \in \mathbb{R}^{C_{out}\times C_{in}\times K_H\times K_W}$ for input X as follows.

$$\tilde{K}_i(X) = K_i \otimes M(X) \quad (4)$$

where $\otimes$ denotes element-wise multiplication. $\tilde{K}_i(X) \in \mathbb{R}^{C_{in}\times K_H\times K_W}$ and $K_i \in \mathbb{R}^{C_{in}\times K_H\times K_W}$ are the *i-th* output channel kernel of $\tilde{K}_i(X)$ and K with $1 \leq i \leq C_{out}$. This proposed ACNN module is applied to the conventional speaker recognition model.

2.2. Speaker recognition model with ACNN

We modified the original ResNet [26] with half of output channels and VGG-M [27], which were widely used in the filed of speaker recognition [28, 29, 30, 31, 32], to **Adaptive Thin ResNet-18** and **Adaptive VGG-M** in which the proposed ACNN module is utilized.

The proposed ACNN module is applied for all convolution layers except the last convolution layer (conv6) which is the frequency pooling layer. The models have a dividing layer that cuts the input according to specific time length W_v and W_r with overlap, respectively, as shown in Table 1. N is the total number of divided segments and they are concatenated on the batch axis. Nonlinear function ReLU and batch normalization are applied after every conv layer (conv1 to conv6). n is the output size of time axis and depends on the variable segment length. The strides or kernels size have been reduced compared to the original networks to obtain short time frame-level features because we want to show the performance according to the various range of W_v and W_r. The time-pooling layer takes a temporal average pooling (TAP) of the features along N and n_c to give equal attention to frame-level features. Since the output before the time pooling layer is a frame-level feature, the networks are determined to have the same number of 512-dimensional frame-level features as $N\times n_c$ for about 3 second input. The utterance-level speaker embedding is a 512-dimensional vector from the fc1 layer result.

3. Experiment Setup and Details

3.1. Input Representations

The spectrograms with 257 frequency bins are generated using a hamming window of width 25ms with step 10ms and number of fast Fourier transform 512. We randomly extract the spectrogram for about 3 seconds from each utterance and use it for training with no data augmentation. Especially, for each model to have a same number of $N \times n_c$ frame-level features as 18, a 257×305 spectrogram of 3.065 seconds is used to Adaptive VGG-M and a 257×295 spectrogram of 2.965 seconds is used to Adaptive Thin ResNet-18. For the test, full spectrogram from each utterance is used. Mean and variance normalization is performed on every frequency bin of the spectrogram.

3.2. Baseline Architecture

To compare the speaker recognition performance, conventional models VGG-M, ResNet-18, ResNet-34, and Thin ResNets with 512-dimensional speaker embeddings were used. Thin ResNets have the half of output channels.

For speaker identification, we train the models using softmax for the number of speakers. For speaker verification, the models are trained using AAM-softmax (ArcFace) [33] with $m = 0.2$ and $s = 30$, which shows better performance and less computation time [31].

3.3. Implementation Details

Our implementation is based on the PyTorch [34] with 4 NVIDIA TITAN RTX GPUs. We use Cross-Entropy loss with Stochastic Gradient Descent (SGD) optimizer with momentum 0.9 and weight decay 5×10^{-4}. An initial learning rate is 10^{-2} decreasing by a factor of 0.95 every epoch. Batch normalization is used with a fixed batch size of 120 and no data augmentation is performed during training.

3.4. Evaluation Metrics

For identification, Top-1 and Top-5 accuracies for the softmax values of 1251 speakers are used as evaluation metrics. For verification, the similarity between the 512-dimensional utterance-level embeddings after training with AAM-softmax is calculated using cosine similarity. We use Equal Error Rate (EER) and the minimum value of the cost function C_{det} as evaluation metrics of verification. EER is the rate at which both acceptance and rejection errors are equal. The C_{det} is a weighted sum of

Table 1: *Adaptive VGG-M and Adaptive Thin ResNet-18 architectures with dividing layer and ACNN module denoting the adaptive convolutional neural network module. Numbers inside parentheses refer to size × size of kernel and # filters. MaxPool is the max pooling layer with size × size, and FC is the fully connected layer with # output nodes. The TAP takes the mean of the features along N and n_c axis. Nonlinear function ReLU and batch normalization are applied after every conv layer.*

Layer	Adaptive VGG-M		Adaptive Thin ResNet-18	
	Structure	**Output shape**	**Structure**	**Output shape**
divide0	-	N×1×257×W_v	-	N×1×257×W_r
conv1	ACNN(7 × 7, 96), stride 2 MaxPool(3 × 3), stride 1 × 2	N×96×125×n_{v1}	ACNN(7 × 7, 32), stride 2 MaxPool(3 × 3), stride 2	N×32×62×n_{r1}
conv2	ACNN(5 × 5, 256), stride 2 MaxPool(3 × 3), stride 2	N×256×30×n_{v2}	[ACNN(3 × 3, 32) ACNN(3 × 3, 32)] × 2, stride 1	N×32×62×n_{r2}
conv3	ACNN(3 × 3, 384), stride 1	N×384 ×30×n_{v3}	[ACNN(3 × 3, 64) ACNN(3 × 3, 64)] × 2, stride 2	N×62×31×n_{r3}
conv4	ACNN(3 × 3, 256), stride 1	N×256×30×n_{v4}	[ACNN(3 × 3, 128) ACNN(3 × 3, 128)] × 2, stride 2	N×128×16×n_{r4}
conv5	ACNN(3 × 3, 256), stride 1 MaxPool(5 × 1), stride 3 × 1	N×256×9×n_{v5}	[ACNN(3 × 3, 256) ACNN(3 × 3, 256)] × 2, stride 2 × 1	N×256×8×n_{r5}
conv6	CNN(9 × 1, 512), stride 1	N×512×1×n_c	CNN(8 × 1, 512), stride 1	N×512×1×n_c
time-pooling	TAP	512	TAP	512
fc1	FC(512)	512	FC(512)	512

false-reject and false-accept error probabilities with parameters $C_{miss} = 1$, $C_{fa} = 1$ and $P_{tar} = 0.01$ [28, 35].

3.5. Dataset

We use the VoxCeleb 1 dataset [28] to train the speaker identification and verification models. It contains total 153,516 utterances for 1,251 speakers. In the speaker identification task, a part of 1,251 speakers' utterances is used to train the model, and the remaining utterances of 1,251 speakers are used for testing. In the speaker verification task, the models are trained using the utterances of 1,211 speakers, and the unseen utterances of 40 speakers with the official trial pairs list of Voxceleb 1 are used to test them [29]. Details about the dataset for speaker identification and verification are shown in Table 2.

Table 2: *Train and test set of Voxceleb 1 for speaker identification and verification.*

Set		# Speakers	# Utterances
Identification	Train	1,251	145,265
	Test	1,251	8,251
Verification	Train	1,211	148,642
	Test	40	4,715

4. Results and Discussion

4.1. Optimal Adaptive Convolutional Neural Network

Proposed ACNN module generates the adaptive kernel from the specific time frame segment, so it is necessary to determine the optimal length of W_v and W_r, which represents the best performance. The input of about 3 seconds has 18 frame-level features when the networks excluding the dividing layer are used. We choose W_v and W_r of the dividing layer with overlaps to determine n_c as 1, 2, 3, 6, and 9. The overlaps of Adaptive VGG-M and Adaptive Thin ResNet-18 are 17 and 7 respectively. Also, we compared the proposed models and the baseline models with the same time-pooling layer and loss to see only the effect of the ACNN in the front-end model. Table 3 shows the speaker identification and verification results.

In the results of the adaptive models, Adaptive VGG-M (Top-1 = 86.51%, EER = 5.68%) and Adaptive Thin ResNet-18 (Top-1 = 85.84%, EER = 6.18%) of $n_c = 1$ show the best performance for text-independent speaker identification and verification. The performance decreases as W increases, but they are still better than the baseline models (VGG-M and Thin ResNet-18). Since the short time segment contains a small number of phonemes (low variance of phonemes) compared to a long segment, it can be inferred that the adaptive kernel tends to be determined according to phonemes, which leads to improvement in overall performance. Thus, the optimal ACNN-based text-independent speaker recognition models with the dividing layer are Adaptive VGG-M with $W_v = 33$ and Adaptive Thin ResNet-18 with $W_r = 23$.

Compared with the baseline models, the performance of the best models has been reliably improved with a parameter increase of approximately 5% only. Also, it shows better performance than the conventional models (Thin ResNet-34, ResNet-18, ResNet-34) with more channels or layer. We can infer that overfitting has occurred in the baseline models due to poor performance of verification tasks that use different speaker pools between train and test set. Therefore, the ACNN-based model not only bypasses the risk of overfitting, but is more effective in terms of parameters and performance than increasing the number of layers and channels in the static model.

4.2. Analysis of Frame-Level Speaker Embeddings

To verify that the model with ACNN computes the accurate frame-level embeddings in the speech for randomly-varying texts with phonemes effectively, we compared the cosine similarity between utterance-level and the frame-level embeddings according to the *phonemes* within a same speaker. Using VGG-M and Adaptive VGG-M with $W = 33$, the frame-level embeddings are extracted from 0.345 seconds length of speech corresponding to 33 frames.

In this experiment, we used TIMIT dataset [36] which has 630 speakers with 10 utterances. Each phoneme was placed in the center of the 0.345 seconds frame to extract the frame-level

Table 3: *Speaker identification (Top-1 and Top-5) and verification (EER and minC_{det}) results according to the networks on VoxCeleb 1 without data augmentation.*

Network	#Parm	W ($N \times n_c$)	Top-1 (%)	Top-5 (%)	EER (%)	min C_{det}
Adaptive VGG-M (proposed)	4.69M	33 (18×1)	**86.51**	**95.31**	**5.68**	**0.510**
	4.69M	49 (9×2)	83.60	94.35	6.25	0.543
	4.70M	65 (6×3)	82.89	94.09	6.65	0.575
	4.73M	113 (3×6)	81.75	93.15	7.16	0.603
	4.77M	161 (2×9)	80.84	92.38	7.97	0.607
VGG-M	4.42M	-	81.08	93.12	7.49	0.610
Adaptive Thin ResNet-18 (proposed)	4.41M	23 (18×1)	**85.84**	**95.29**	**6.18**	0.589
	4.41M	39 (9×2)	84.61	94.99	6.50	**0.567**
	4.42M	55 (6×3)	83.98	94.51	7.07	0.604
	4.46M	103 (3×6)	83.58	93.42	7.63	0.639
	4.51M	151 (2×9)	81.66	92.99	8.99	0.643
Thin ResNet-18	4.11M	-	81.38	94.26	9.79	0.686
Thin ResNet-34	6.64M	-	83.32	94.26	9.79	0.686
ResNet-18	13.53M	-	84.95	94.81	**8.50**	**0.622**
ResNet-34	23.63M	-	**85.54**	**95.30**	9.13	0.693

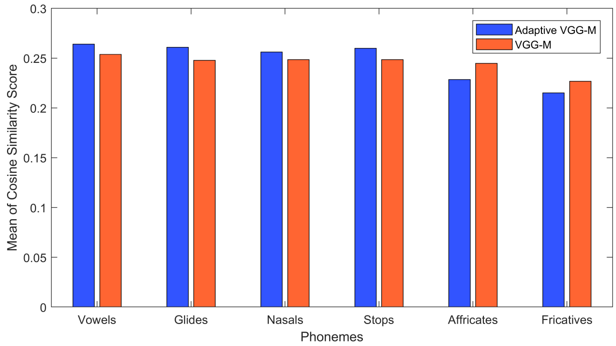
embeddings. For enrollment of each speaker, the utterance-level embedding is derived by averaging the 9 utterance-level embeddings, excluding the target utterance from which frame-level embeddings are extracted [7]. A total of 52 phonemes are classified into 6 categories of stops, affricates, fricatives, nasals, glides, and vowels. For each category, the mean and standard deviation of cosine similarity score are calculated and compared as shown in Figure 2.

Adaptive VGG-M has higher average similarity score than VGG-M in vowels, glides, nasals, and stops. It can be seen that the proposed model using ACNN extracted accurate frame-level embeddings including more speaker information from these phonemes. Also, the model was trained itself to concentrate on vowels and nasals, which have a major influence on speaker recognition performance [7, 8, 9], rather than fricatives. In addition, for the four phonemes with a high average similarity score, the standard deviations of Adaptive VGG-M are lower than that of VGG-M, especially nasals. The variance of embeddings between phonemes within each category has decreased, and it means that frame-level speaker embeddings are extracted phoneme-independently. However, the cosine similarity decreased for affricates and fricatives. We can infer that this is the result of insufficient speaker information because the fricatives is generated by turbulent flow, not in the speaker's own organ.

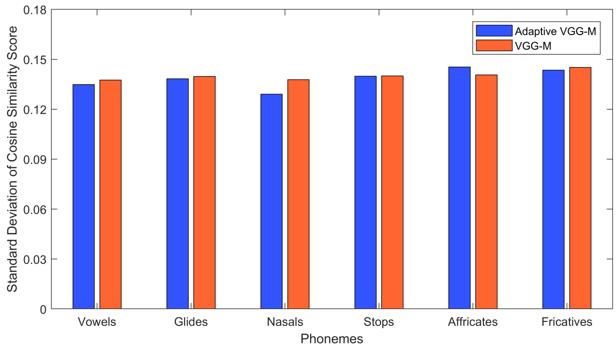
Therefore, it was confirmed that the ACNN-based model extracts the relatively accurate speaker embedding according to characteristics of phonemes, especially vowels and nasal sounds. This result verifies the initial assumption, and it is appropriate to apply the ACNN-based model to text-independent speaker recognition.

5. Conclusion

In this paper, we proposed a text-independent speaker recognition model applying the ACNN module for text-varying speech. The utterances were divided into short time segments and they were applied to the model to determine the adaptive kernel along the time axis by reflecting the characteristics of the speech whose content varies over time according to its phonemes.



(a) *Mean of cosine similarity score*



(b) *Standard Deviation of cosine similarity score*

Figure 2: *Mean and standard deviation of cosine similarity score according to phoneme categories.*

Adaptive VGG-M with $W = 33$ showed the best performance in text-independent speaker recognition. The ACNN-based model outperformed the static model with increased layers and channels by avoiding overfitting, and achieve better performance with smaller model size. Also, the suitable kernels are determined for each short segment, and it effectively extracts accurate speaker information according to characteristics of phonemes, especially with vowels and nasals. The extraction of accurate frame-level embedding leads to accurate utterance level embedding, which shows better performance in text-independent speaker recognition of text-varying speech.

The ACNN-based model and its analysis provide insight into the direction of text-independent speaker recognition that it should consider the speaker features adaptive to the frame-level segment (phonemes, etc.) rather than the entire utterance. As a starting point of this insight, we applied ACNN to text-independent speaker recognition, and it is necessary to check whether the same tendency is observed in other baseline models with ACNN. Furthermore, we will examine the effect of the adaptive kernel over time-frequency and smaller segment lengths in future works.

6. Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00162, Development of Human-care Robot Technology for Aging Society) and Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) funded by the Ministry of Trade, Industry & Energy, Republic of Korea (Grant No. 20204030200050).

7. References

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

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

[3] 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, Conference Proceedings, pp. 5115–5119.

[4] J. Rohdin, A. Silnova, M. Diez, O. Plchot, P. Matějka, and L. Burget, "End-to-end dnn based speaker recognition inspired by i-vector and plda," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, Conference Proceedings, pp. 4874–4878.

[5] 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 *INTERSPEECH*, 2019, Conference Proceedings, pp. 361–365.

[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, Conference Proceedings, pp. 5329–5333.

[7] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, Conference Proceedings, pp. 1007–1013.

[8] J. P. Eatock and J. S. Mason, "A quantitative assessment of the relative speaker discriminating properties of phonemes," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 1994, Conference Proceedings, pp. I/133–I/136 vol. 1.

[9] C.-S. Jung, M. Y. Kim, and H.-G. Kang, "Selecting feature frames for automatic speaker recognition using mutual information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1332–1340, 2009.

[10] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *arXiv preprint arXiv:1807.06514*, 2018.

[11] 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, Conference Proceedings, pp. 3–19.

[12] N. N. An, N. Q. Thanh, and Y. Liu, "Deep cnns with self-attention for speaker identification," *IEEE Access*, vol. 7, pp. 85 327–85 337, 2019.

[13] P. Safari and J. Hernando, "Self-attention encoding and pooling for speaker recognition," *arXiv preprint arXiv:2008.01077*, 2020.

[14] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Interspeech*, 2018, Conference Proceedings, pp. 3573–3577.

[15] A. Hajavi and A. Etemad, "Knowing what to listen to: Early attention for deep speech representation learning," *arXiv preprint arXiv:2009.01822*, 2020.

[16] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[17] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Advances in neural information processing systems*, 2016, Conference Proceedings, pp. 667–675.

[18] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," in *Advances in Neural Information Processing Systems*, 2019, Conference Proceedings, pp. 1307–1318.

[19] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, Conference Proceedings, pp. 11 030–11 039.

[20] C. Chen and Q. Ling, "Adaptive convolution for object detection," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3205–3217, 2019.

[21] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, "Pixel-adaptive convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 11 166–11 175.

[22] J. Zamora Esquivel, A. Cruz Vargas, P. Lopez Meyer, and O. Tickoo, "Adaptive convolutional kernels," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, Conference Proceedings.

[23] D. Shen, M. R. Min, Y. Li, and L. Carin, "Learning context-sensitive convolutional filters for text processing," *arXiv preprint arXiv:1709.08294*, 2017.

[24] B.-J. Choi, J.-H. Park, and S. Lee, "Adaptive convolution for text classification," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, Conference Proceedings, pp. 2475–2485.

[25] B. Gu, W. Guo, L. Dai, and J. Du, "An adaptive x-vector model for text-independent speaker verification," *arXiv preprint arXiv:2002.06049*, 2020.

[26] 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, Conference Proceedings, pp. 770–778.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[29] 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.

[30] J. S. Chung, J. Huh, and S. Mun, "Delving into voxceleb: environment invariant speaker recognition," *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pp. 349–356, 2020.

[31] 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," *arXiv preprint arXiv:2003.11982*, 2020.

[32] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

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

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, Conference Proceedings, pp. 8026–8037.

[35] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The Speakers in the Wild (SITW) speaker recognition database," in *Interspeech*, 2016, Conference Proceedings, pp. 818–822.

[36] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, p. 27403, 1993.