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

Shallow Convolution-Augmented Transformer with Differentiable Neural Computer for Low-Complexity Classification of Variable-Length Acoustic Scene

Soonshin Seo, Donghyun Lee, and Ji-Hwan Kim†

Dept. of Computer Science and Engineering, Sogang University, Seoul, Republic of Korea
{ssseo, redizard, kimjihwan}@sogang.ac.kr

Abstract

Convolutional neural networks (CNNs) exhibit good performance in low-complexity classification with fixed-length acoustic scenes. However, previous studies have not considered variable-length acoustic scenes in which performance degradation is prevalent. In this regard, we investigate two novel architectures—convolution-augmented transformer (Conformer) and differentiable neural computer (DNC). Both the models show desirable performance for variable-length data but require a large amount of data. In other words, small amounts of data, such as those from acoustic scenes, lead to overfitting in these models. In this paper, we propose a shallow convolution-augmented Transformer with a differentiable neural computer (shallow Conformer-DNC) for the low-complexity classification of variable-length acoustic scenes. The shallow Conformer-DNC is enabled to converge with small amounts of data. Short-term and long-term contexts of variable-length acoustic scenes are trained by using the shallow Conformer and shallow DNC, respectively. The experiments were conducted for variable-length conditions using the TAU Urban Acoustic Scenes 2020 Mobile dataset. As a result, a peak accuracy of 61.25% was confirmed for shallow Conformer-DNC with a model parameter of 34 K. It is comparable performance to state-of-the-art CNNs.

Index Terms: acoustic scene classification, low-complexity, variable-length, Conformer, differentiable neural computer

1. Introduction

Acoustic scene classification (ASC) is the task of predicting specific locations of sound events and auditory information [1]. Recent studies have shown that the performance of ASCs has dramatically improved owing to the emergence and use of deep learning technology. Convolutional neural networks (CNNs) have mainly been used to convert acoustic signals into spectrograms and then for training them [2–6]. CNNs can learn the correlations of local information of the input. In particular, the appearance of residual learning has the training of CNNs without increasing parameters [7]. These residual CNNs greatly contributed to the performance improvement of ASCs [8–11].

McDonnell *et al*. proposed residual CNNs using spectrogram separation to the classification of fixed-length acoustic scenes recorded using multiple devices [10]. The frequency domains of the spectrogram were separated in half for training the frequency response of each device. The generated high-frequency spectrogram and low-frequency spectrogram were then trained using each residual CNN. It showed generalization performance for test cases recorded with unknown devices. However, this study did not cover the low-complexity and variable-length issues.

Hu *et al*. proposed MobNet and small-FCNN for low-complexity classification of fixed-length acoustic scenes [11]. Data augmentation methods such as mixup [12] and spectrum augmentation [13] were also used in the training. The MobNet is a residual CNN derived from MobileNetV2 [14], which has the low-complexity and high accuracy. Small-FCNN is also a residual CNN that uses fully connected layers and channel attention. These studies showed that the models exhibited good performance under low-complexity conditions, but did not cover the variable-length issue.

In the ASC under low-complexity conditions, the variable-length issue disturbs the generalization performance. In particular, the performance of test cases with long-length data is degraded for models trained with short-length data. In this regard, we focus on two novel architectures—convolution-augmented Transformer (Conformer) [15], and differentiable neural computer (DNC) [16]. The Conformer is a model in which the CNN and Transformer [17] are combined. It has been exhibited good performance in automatic speech recognition and continuous speech separation for variable-length [15,18]. The Conformer can learn the local and global dependencies of an acoustic signal simultaneously by using convolution and self-attention mechanisms. Moreover, the Conformer has the advantage of training in the short-term context. However, the longer the length of the training data, the lower is the efficiency of self-attention.

The DNC can be used to compensate for the limitations of the Conformer. The DNC showed performance improvement compared to recurrent neural networks (RNNs) in the inference task for variable-length [16,19–22]. The DNC consists of a controller and an external memory. Since DNC can store long sequence information to external memory, it has the advantage of training the long-term context. However, it is difficult to train high-complexity models such as the Conformer and DNC using small amounts of data. The acoustic scene data are mostly composed of small amounts of less than 100 hours [23,24]. Since these data are not enough for training both models, an overfitting problem can occur.

In this paper, we propose a shallow convolution-augmented Transformer with a differentiable neural computer (shallow Conformer-DNC) for the low-complexity classification of variable-length acoustic scenes. The previous Conformer and DNC are modified for training with small amounts of acoustic scene data. Then, the two models were combined into a one-pass training. The proposed shallow Conformer-DNC can simultaneously learn the short-term and long-term context of variable-length acoustic scenes.

We will introduce the previous Conformer and DNC in Section 2. In Section 3, we describe the proposed shallow

† Corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-1308

Conformer-DNC and show the results in Section 4. Finally, conclusions are made in Section 5.

2. Previous Works

2.1. Conformer

The Conformer is one of the powerful deep learning models using convolution and self-attention mechanisms [15]. As shown in Figure 1, the Conformer consists of several pre-processing layers and N of Conformer blocks. The subsampled feature is generated using several pre-processing layers—SpecAugment [13], convolution subsampling, linear, and dropout.

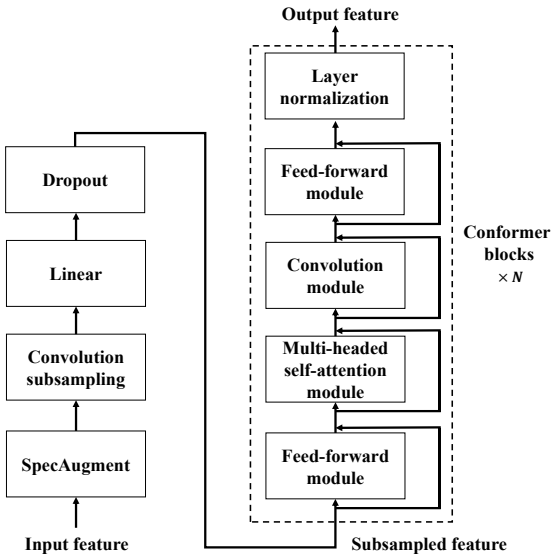

Figure 1: *Structural overview of the Conformer.*

Then, the feature is fed into the Conformer blocks. The Conformer block consists of two feed-forward modules, a multi-headed self-attention module, a convolution module, and layer normalization. Residual connections are applied between each module. Then, the output feature is generated by using layer normalization.

2.2. Differentiable neural computer

The DNC is one of the memory-augmented deep learning models using an attention mechanism [16]. Figure 2 is a structural overview of the DNC, which consists of a controller and an external memory. The controller is a deep learning model, such as RNN or CNN. The external memory is determined by the number of memory addresses and the dimensionality of the memory vector.

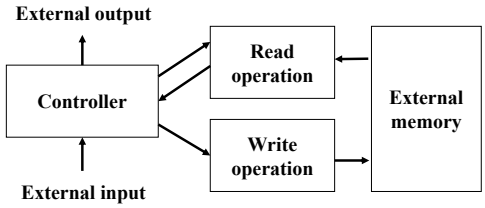

Figure 2: *Structural overview of the DNC.*

An external input is used as an input of the controller. An interface vector and a controller output vector are generated from the controller. The interface vector determines an index of the external memory address accessed at time step t to perform the read and write operation. The controller output vector is equal to the output vector of the highest hidden layer in the controller. After the read and write operation, read vectors are generated. Read vectors are generated with the attention mechanism. These vectors are projected into the dimension of the controller output vector. An external output is generated from the addition of projected read vectors and the controller output vector.

3. Shallow Convolution-Augmented Transformer with Differentiable Neural Computer

3.1. Shallow Conformer

The proposed shallow Conformer is derived from the previous Conformer encoder [15]. First, a sub-sampling block is applied to the variable-length input feature $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \dots, \boldsymbol{f}_l, \dots, \boldsymbol{f}_L\}$ $(\boldsymbol{f}_l \in \mathbb{R}^d)$ of length L, as shown in Figure 3. The sub-sampling block consists of convolution subsampling, linear transformation, and dropout. In the convolution subsampling operation, the quarter areas of the input feature are extracted by using two convolution and nonlinear activations. Then, linear transformation and dropout regularization are applied to subsampled feature $\boldsymbol{SF} = \{\boldsymbol{sf}_1, \boldsymbol{sf}_2, \dots, \boldsymbol{sf}_l, \dots, \boldsymbol{sf}_{4/L}\}$ $(\boldsymbol{f}_l \in \mathbb{R}^d)$.

Then, subsampled feature $\boldsymbol{SF}$ is fed into the several Conformer blocks (in this paper, we fix the number of Conformer blocks to two). The Conformer blocks consist of two feed-forward modules, a multi-headed self-attention module, a convolution module, and layer normalization. Except for layer normalization, each module is connected residually. In particular, half weights of the feed-forward module are used in the residual connections.

In the Conformer block, the feature $\boldsymbol{X}$ is generated passing through using the first feed-forward module. After the application of pre-normalization to input $\boldsymbol{X}$, linear transformation and swish activation [25] are applied to it. Following this, two dropouts and linear transformations are applied. Then, feature $\dot{\boldsymbol{X}}$ is generated after passing through the multi-headed self-attention module. Relative positional encoding is then applied to generalize the various input lengths for self-attention [15]. After this, dropout is applied for regularization.

Next, feature $\ddot{\boldsymbol{X}}$ is generated through the convolutional module. After applying pre-normalization to input $\dot{\boldsymbol{X}}$, point-wise convolution and gated linear unit activation (GLU) [26] are applied. Then, 1D-depthwise convolution and swish activation are applied, which is followed by second point-wise convolution and dropout. Then, after passing through the second feed-forward module, layer normalization is applied, and the output feature $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_t, \dots, \boldsymbol{y}_T\}$ $(\boldsymbol{y}_t \in \mathbb{R}^D)$ of length T, are generated. The above process can be mathematically summarized as follows:

$$\dot{\boldsymbol{X}} = \boldsymbol{X} + 1/2\ feed\_forward(\boldsymbol{X}) \quad (1)$$

$$\ddot{\boldsymbol{X}} = \dot{\boldsymbol{X}} + multi\_headed\_self\_attention(\dot{\boldsymbol{X}}) \quad (2)$$

$$\ddot{\boldsymbol{X}} = \ddot{\boldsymbol{X}} + convolution(\ddot{\boldsymbol{X}}) \quad (3)$$

$$\boldsymbol{Y} = norm(\ddot{\boldsymbol{X}} + 1/2\ feed\_forward(\ddot{\boldsymbol{X}}) \quad (4)$$

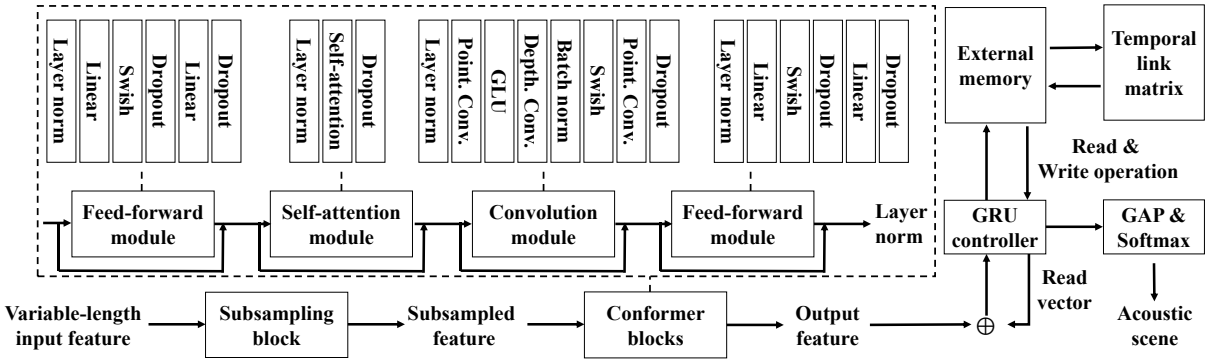

Figure 3: *Structural overview of the proposed shallow Conformer-DNC.*

3.2. Shallow differentiable neural computer

The output feature $\boldsymbol{Y}$ generated by the shallow Conformer is used as an input vector to the shallow DNC. The proposed shallow DNC consists of a gated recurrent unit (GRU)-based controller and external memory. Its external memory can be represented as $\mathbb{R}^{A\times D}$, where A is the number of memory addresses, and D is the dimensionality of the memory vector. An external input is a concatenated vector containing an input vector $\boldsymbol{y}_t$ at time step t and read vectors $\boldsymbol{rv}_{t-1}^{i}$ at time step $t-1$ $(i = 1,2,\dots,R)$. The $\boldsymbol{rv}_t^i$ was generated using a read operation. It is defined as

$$\boldsymbol{rv}_t^i = M_t^{\mathrm{T}}\boldsymbol{aw}_t^{r,i}, \tag{5}$$

where M_t^{T} and $\boldsymbol{aw}_t^{r,i}$ are the transposed external memory and i-th read attention weighting vector at time step t, respectively. Before the read operation, the controller performs the write operation, which is defined as

$$M_t = M_{t-1}\circ\left(OM - \boldsymbol{aw}_t^{w}\boldsymbol{ev}_t^{\mathrm{T}}\right) + \boldsymbol{aw}_t^{w}\boldsymbol{ci}_t^{\mathrm{T}}, \tag{6}$$

where OM is an $A \times D$ matrix with all elements equal to one. $\boldsymbol{aw}_t^{w}$ is the write attention weighting vector at time step t, $\boldsymbol{ev}_t^{\mathrm{T}}$ is the transposed erase vector at time step t, and $\boldsymbol{ci}_t^{\mathrm{T}}$ is the transposed converted external input at time t.

To generate $\boldsymbol{aw}_t^{r,i}$, content-based and temporal linking addressing was used [21]. The content-based addressing calculates the cosine similarity between every memory vector and a key vector generated by the controller. Temporal linking addressing uses a temporal link matrix to determine the memory vector to be written after or before the read operation in the previous time step. To generate $\boldsymbol{aw}_t^{w}$, content-based addressing and memory-allocation-based addressing have been used [21]. Memory-allocation-based addressing uses usage vectors to determine the degree of usage in each memory vector.

Then, read vectors are transformed to the dimension of the controller output vector. An external output is generated from the element-wise additions of transformed read vectors and the controller output vector. Finally, the external output of the DNC is transformed to the number of classes using global average pooling (GAP), and softmax is applied.

4. Experiments

4.1. Dataset

We used the development dataset of TAU Urban Acoustic Scenes 2020 Mobile [24]. (The evaluation dataset was not published). The recordings in the dataset were collected from 10 acoustic locations—airport, indoor shopping mall, metro station, pedestrian street, public square, street with a medium level of traffic, traveling by tram, traveling by bus, traveling by an underground metro, and an urban park. Also, 3 real devices and 6 simulated devices were used for the dataset.

The total number of recordings is 23,040. The dataset is split into training and test set with a 70% ratio as a cross-validation setup (some recordings are not used for training/test split). The number of training and test set is 13,962 and 2,970, respectively. The recordings using 3 simulated devices are included only in the test set. The duration of each training and test recordings is fixed at 10 sec. The sampling rates are fixed at 44.1 kHz, 24-bit resolution, and mono channel.

4.2. Experimental configurations

Table 1: *Model hyperparameters for the proposed shallow Conformer-DNC (C, D, and EM refer to Conformer, DNC, and external memory, respectively).*

Description	Shallow Conformer-DNC (S)	Shallow Conformer-DNC (M)	Shallow Conformer-DNC (L)
Params (K)	34.0	75.1	132.3
C-Blocks	2	2	2
C-Dim	16	24	32
C-Att. Heads	4	4	4
C-Conv. Kernels	7	13	19
D-Layers	1	1	1
D-GRU-Dim	16	24	32
D-EM-Addresses	16	24	32
D-EM-Dim	16	24	32

For each input acoustic signal, a short-time Fourier transform with 2048 FFT points was performed with a hop length of 1024 samples. We then extracted a 128-dimensional log-Mel spectrogram. The number of time bins varied according to the duration of the input acoustic signal, and a feature of $128 \times the\ number\ of\ time\ bins \times 1$ was generated (i.e., for 10 sec, a feature of $128 \times 431 \times 1$ was generated.)

We trained the proposed models using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, an initial learning rate of 10^{-2}, and a weight decay of 10^{-6}. Also, we used two types of learning rate scheduler; one was a transformer learning rate scheduler [15] with 800 warm-up steps and a peak learning rate of $0.05/\sqrt{d}$. (d is the dimension of the Conformer), and the other was a learning rate scheduler [27] that reduced the learning rate in the case of validation loss plateaus. We also

trained the proposed model using a standard cross-entropy loss function with a batch size of 512 and epoch 200.

We designed three models with different hyperparameters, as presented in Table 1. Depending on the model parameter size, the models were defined as small (S), medium (M), and large (L). Among them, the small model with the lowest model parameter was mainly used for the experiments.

4.3. Experimental results

Four experiments were performed using the proposed model. In the first experiment, the performance of the proposed models using three hyperparameters was evaluated and the results are presented in Table 2. As can be seen from the table, the shallow Conformer (S) showed 52.56% accuracy with 24.1 K model parameters, whereas the shallow Conformer-DNC (S) showed 52.73% accuracy with 34 K model parameters.

Table 2: *Experimental results on proposed models using three hyperparameters.*

Model	Params (K)	Acc (%)
Shallow Conformer (S)	24.1	52.56
Shallow Conformer (M)	53.3	54.68
Shallow Conformer (L)	94.0	53.17
Shallow Conformer-DNC (S)	34.0	52.73
Shallow Conformer-DNC (M)	75.1	54.72
Shallow Conformer-DNC (L)	132.3	55.29

For the second experiment, normalization and data augmentation were applied to the proposed model, as detailed in Table 3. Normalization was performed using zero mean and unit variances for each frequency bin of the log-Mel spectrogram [28]. Also, mixup and spectrum augmentation were used in the training [11]. Based on the results of the experiment, it was confirmed that the use of spectrum augmentation showed an absolute performance difference of 6.8%.

Table 3: *Ablation study of the proposed model using normalization and data augmentations.*

Model	Params (K)	Acc (%)
Shallow Conformer-DNC (S)	34.0	61.25
w/o normalization		55.05
w/o mixup ($\alpha = 0.2$)		59.33
w/o spectrum augmentation		54.45

In the third experiment, the performance of the proposed model was compared with that of the state-of-the-art CNNs, as presented in Table 4. All the models applied the same normalization and data augmentation techniques, as described in Table 3. Based on the results of the experiment, it was confirmed that the proposed model showed better performance than Residual CNN [10] and Small-FCNN [11].

In the last experiment, the robustness of the proposed models and state-of-the-art CNNs under variable-length conditions was evaluated, and the results are presented in Table 5. The training and testing were conducted using recordings with lengths ranging from 1 to 10 sec. Based on the results of the experiment, it was confirmed that the proposed shallow Conformers outperformed the state-of-the-art CNNs when training using a short-length (i.e. 1 to 3 sec). Also, the proposed shallow Conformer-DNC achieved a peak accuracy of 61.25% with a model parameter of 34 K. This performance is comparable to those of state-of-the-art CNNs with accuracies ranging from 51.25% to 61.83% and model parameters ranging from 34.4 K to 35 K.

Table 4: *Performance comparison between the proposed model and state-of-the-art CNNs*.

Model	Params (K)	Acc (%)
DCASE 2021 task 1a baseline	46.2	46.40
Residual CNN [10] # num. of stacks & filters = 2 & 10	34.4	51.25
MobNet [11] # num. of filters = {10, 14, 18}	35.0	**61.83**
Small-FCNN [11] # num. of filters = {14, 26, 38}	34.5	56.64
Shallow Conformer-DNC (S)	**34.0**	**61.25**

Table 5: *Robustness test under variable-length conditions for the proposed models and state-of-the-art CNNs (TRL: training recordings length).*

Model	TRL	Test recordings length					
		1s	3s	5s	7s	9s	10s
Residual CNN [10]	1s	37.47	35.34	34.27	31.87	30.86	30.49
	3s	33.79	44.31	45.52	47.24	46.26	37.04
	5s	29.21	42.89	46.87	47.78	48.38	48.62
	7s	27.73	45.55	45.55	48.52	49.43	50.20
	9s	27.80	44.64	44.64	48.11	50.40	50.67
	10s	21.60	45.86	45.86	48.35	50.57	51.25
MobNet [11]	1s	35.44	34.10	35.28	34.33	33.22	32.28
	3s	26.99	45.22	47.94	51.35	52.70	52.70
	5s	29.28	47.04	52.96	53.98	56.84	55.66
	7s	29.21	46.73	54.35	**58.02**	**59.33**	**59.80**
	9s	28.13	**48.11**	**54.35**	**59.23**	**61.46**	**61.42**
	10s	27.16	**45.79**	**53.27**	**57.72**	**60.88**	**61.83**
Small-FCNN [11]	1s	34.80	36.35	34.70	34.00	32.88	32.14
	3s	**36.73**	47.98	50.44	50.94	51.08	51.28
	5s	**33.93**	**47.61**	**54.11**	**55.80**	57.21	58.15
	7s	32.04	**49.29**	**54.14**	56.97	58.39	58.89
	9s	**30.53**	46.66	52.09	54.04	55.86	56.57
	10s	28.81	44.51	49.97	52.96	55.26	56.64
Shallow Conformer (S)	1s	**38.11**	**39.96**	**39.86**	**39.76**	**40.87**	**40.77**
	3s	35.98	**49.70**	**51.08**	**54.89**	**55.49**	**55.83**
	5s	30.83	45.82	51.25	55.22	56.97	58.12
	7s	29.55	44.14	51.52	55.56	57.48	59.23
	9s	28.87	43.16	50.24	54.92	58.46	58.36
	10s	25.10	42.52	50.98	55.56	58.69	59.77
Shallow Conformer-DNC (S)	1s	36.32	37.30	37.84	37.53	36.59	36.59
	3s	34.84	48.05	50.98	53.27	53.57	53.54
	5s	28.54	45.49	52.70	54.82	**57.95**	**58.46**
	7s	**32.14**	44.95	51.55	56.00	57.92	58.79
	9s	29.14	44.78	52.36	55.86	58.05	58.22
	10s	27.90	43.94	51.79	56.37	59.47	61.25

5. Conclusions

We proposed a shallow Conformer-DNC for the low-complexity classification of variable-length acoustic scenes. The proposed shallow Conformer-DNC can learn short-term and long-term contexts of variable-length acoustic scenes simultaneously. Based on the results of variable-length condition test, the results obtained are comparable to those of state-of-the-art CNNs.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2020R1F1A1076562).

7. References

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[2] S. S. R. Phaye, E. Benetos, and Y. Wang, "SubSpectralNet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *Proceedings ICASSP 2019 – 44th IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, May. 2019, pp. 825–829.

[3] J. Jung, H. Shim, J. Kim, S. Kim, and H. Yu, "Acoustic scene classification using audio tagging," in *Proceedings INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, Oct. 2020, pp. 1176–1180.

[4] L. Zhang, J. Han, and Z. Shi, "ATReSN-Net: capturing attentive temporal relations in semantic neighborhood for acoustic scene classification," in *Proceedings INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, Oct. 2020, pp. 1181–1185.

[5] D. V. Devalraju, M. H, P. Rajan, and D. A. Dinesh, "Attention-driven projections for soundscape classification," in *Proceedings INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, Oct. 2020, pp. 1206–1210.

[6] Z. Kwiatkowska, B. Kalinowski, M. Kośmider, and K. Rykaczewski, "Deep learning based open set acoustic scene classification," in *Proceedings INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, Oct. 2020, pp. 1216–1220.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings CVPR 2016 – 29th IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, Jun./Jul. 2016, pp. 770–778.

[8] L. Ford, H. Tang, F. Grondin, and J. R. Glass, "A deep residual network for large-scale acoustic scene analysis," in *Proceedings INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, Graz, Austria, Sem. 2019, pp. 2568–2572.

[9] J. Naranjo-Alcazar, S, Perez-Castanos, P, Zuccarello, and M. Cobos, "Acoustic scene classification with squeeze-excitation residual networks," *IEEE Access*, vol. 8, pp. 112287–112296.

[10] M. D. McDonnell, and W. Gao, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," in *Proceedings ICASSP 2020 – 45th IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, May. 2020, pp. 141–145.

[11] H. Hu, C. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, J. Du, and C. Lee, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," *arXiv preprint arXiv:2007.08389*, 2020.

[12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[13] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: a simple data augmentation method for automatic speech recognition," in *Proceedings INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, Graz, Austria, Sem. 2019, pp. 2019–2680.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings CVPR 2018 – 31st IEEE conference on computer vision and pattern recognition*, Salt Lake City, USA, Jun. 2016, pp. 4510–4520.

[15] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: convolution-augmented transformer for speech recognition," in *Proceedings INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, Oct. 2020, pp. 5036–5040.

[16] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, and D. Hassabis. "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings NeurIPS 2017 – 31st Conference on Neural Information Processing Systems*, Long Beach, USA, Dec. 2017, pp. 6000–6010.

[18] S. Chen, Y. Wu, Z. Chen, J. Li, C. Wang, S. Liu, and M. Zhou, "Continuous speech separation with conformer," *arXiv preprint arXiv:2008.05773*, 2020.

[19] R. Csordás, and J. Schmidhuber, "Improving differentiable neural computers through memory masking, de-allocation, and link distribution sharpness control," *arXiv preprint arXiv:1904.10278*, 2019.

[20] W. Luo, and F. Yu, "Recurrent highway networks with grouped auxiliary memory," *IEEE Access*, vol. 7, pp. 182037–182049, 2019.

[21] D. Lee, H. Park, S. Seo, H. Son, G. Kim, and J. Kim, "Robustness of differentiable neural computer using limited retention vector-based memory deallocation in language model," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 3, pp. 837–852, 2021.

[22] D. Lee, H. Park, S. Seo, C. Kim, H. Son, G. Kim, and J. Kim, "Language model using differentiable neural computer based on forget gate-based memory deallocation," *Computers, Materials & Continua*, vol. 68, no.1, pp. 537–551, 2021.

[23] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings DCASE Workshop 2018 – 3rd Workshop on Detection and Classification of Acoustic Scenes and Events*, Surrey, UK, Nov. 2018, pp. 9–13.

[24] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings DCASE Workshop 2020 – 5th Workshop on Detection and Classification of Acoustic Scenes and Events*, Tokyo, Japan, Nov. 2020, pp. 56–60.

[25] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[26] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings ICML 2017 – 34th International Conference on Machin Learning*, Sydney, Australia, Aug. 2017, pp. 933–941.

[27] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, and J. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system. in *Proceedings INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, Graz, Austria, Sem. 2019, pp. 2928–2932.

[28] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," in *Proceedings DCASE Workshop 2020 – 5th Workshop on Detection and Classification of Acoustic Scenes and Events*, Tokyo, Japan, Nov. 2020, pp. 100–104.