IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.7 JULY 2024

LETTER

Dual-Path Convolutional Neural Network Based on Band Interaction Block for Acoustic Scene Classification

Pengxu JIANG[†a)], **Yang YANG**[†], *Nonmembers*, **Yue XIE**[††], *Member*, **Cairong ZOU**[†], *and* **Qingyun WANG**[††], *Nonmembers*

SUMMARY Convolutional neural network (CNN) is widely used in acoustic scene classification (ASC) tasks. In most cases, local convolution is utilized to gather time-frequency information between spectrum nodes. It is challenging to adequately express the non-local link between frequency domains in a finite convolution region. In this paper, we propose a dual-path convolutional neural network based on band interaction block (DCNN-bi) for ASC, with mel-spectrogram as the model's input. We build two parallel CNN paths to learn the high-frequency and low-frequency components of the input feature. Additionally, we have created three band interaction blocks (bi-blocks) to explore the pertinent nodes between various frequency bands, which are connected between two paths. Combining the time-frequency information from two paths, the bi-blocks with three distinct designs acquire non-local information and send it back to the respective paths. The experimental results indicate that the utilization of the bi-block has the potential to improve the initial performance of the CNN substantially. Specifically, when applied to the DCASE 2018 and DCASE 2020 datasets, the CNN exhibited performance improvements of 1.79% and 3.06%, respectively.

key words: *acoustic scene classification, convolutional neural networks, band interaction block, mel-spectrogram*

1. Introduction

Acoustic scene classification (ASC) uses intelligent devices to judge the sound environment, which has applications in many fields [1]. Scholars based on acoustic scene classification mainly focus on the Detection and Classification of Acoustic Scenes and Events (DCASE). Early research based on ASC focused on the extraction of hand-crafted features. However, it is difficult to avoid the artificial selection of features for hand-crafted features, which will lead to the limited performance of ASC-based systems [2].

Most academics now use convolutional neural networks (CNNs) to categorize the acoustic environment. Spectrum is often the input for CNN-based algorithms. To enhance the performance of the model, some studies [3], [4] combine the first and second derivatives of the spectrum. To further explore the nature of spectrogram, some work [4], [5] cutting the frequency axis from the spectrum and entering it into CNN, it was discovered that this might enhance ASC's performance. Additionally, computer vision and natural language processing frequently employ attention mechanisms [6], [7]. According to the study on the attention module built on the ASC, it can also enhance CNN's functionality.

Even though the aforementioned work has been successfully implemented in ASC, several issues still need to be resolved. First off, various frequency bands need to be addressed differently since the high-frequency and low-frequency components of the scene audio may have distinct physical meanings. Previous research created parallel CNN paths for several frequency bands. This paradigm is intended to make it easier to gather scene data in various frequency bands. However, feature cutting will lead to incomplete information acquisition of different branches at the cutting point, that is, loss of some frequency information. Furthermore, frequency cascades are a common feature of late fusion based on many frequency bands. We have observed that attention mechanism, such as the RNN-LSTM model based on attention mechanism [8], has outstanding effects in the feature fusion stage. It is also essential to consider attention-based late fusion mode for the frequency band.

In view of this, this paper has designed a Dual-path convolutional neural network based on band interaction block (DCNN-bi) for ASC, and the designed model is shown in Fig. 1. Inspired by some previous work [4], [5], mel-spectrograms and their first and second derivatives serve as the model's input, and these features are further separated into high-frequency portions and low-frequency parts to input two parallel CNNs to accommodate the demands of the model. Additionally, we have designed three band interaction blocks (bi-blocks) with various structural that are positioned at the late feature fusion stage and the convolution stage of different paths, respectively. The bi-block's primary objective is to gather additional time-frequency information by combining local information from the two paths. The bi-block α, which serves as the area where the two paths' information interact, offers a full time-frequency receptive field to aid in collecting time-frequency-related information at frequency truncation in various frequency bands. The bi-block β and bi-block γ are set at the late fusion stage, focusing on acquiring the attention-based non-local weight and channel weight rather than exploring the time-frequency relationship between individual dimensions. The experiment discusses the improvement of different modules on the baseline CNN performance and verifies the superiority of the proposed module through the two related databases.

Manuscript received June 29, 2023.
Manuscript revised August 23, 2023.
Manuscript publicized October 4, 2023.

[†]The authors are with the School of Information Science and Engineering, Southeast University, P.R. China.
[††]The authors are with School of Communication Engineering, Nanjing Institute of Technology, P.R. China.
a) E-mail: 230209051@seu.edu.cn

DOI: 10.1587/transfun.2023EAL2056

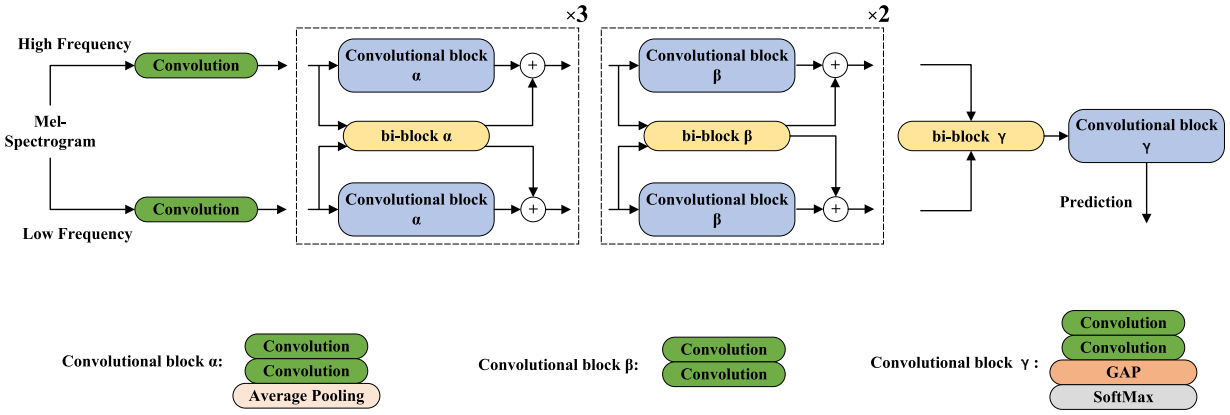

Fig. 1 Illustration of the proposed DCNN-bi.

2. Methods

2.1 Input Feature

We use mel-spectrogram as the input feature, the same as most CNN-based studies [9]–[11], and mel-spectrogram is a two-dimensional feature with frequency and time axes. In addition, we calculate the first and second derivatives of spectrograms to form three-dimensional features, similar to RBG image representation. Which is expressed as:

$$\chi \in \mathbb{R}^{F\times T\times C}, \tag{1}$$

where F, T, and C represent frequency, time, and channels, respectively. In image recognition, the object's location does not influence the model's ability to discriminate. However, high-frequency and low-frequency features in the mel-spectrogram may have different meanings. Therefore, our ASC system has two input paths. The model's inputs are the high-frequency and low-frequency components.

2.2 Convolutional Block

We have designed three convolution modules for CNN: Convolutional block α, Convolutional block β, and Convolutional block γ. Figure 1 depicts the convolution network's structural layout.

Two convolutional layers and one average pooling layer comprise the convolutional block α. Only two convolutional layers are present in convolutional block β. Two layers of convolution, one layer of global average pooling (GAP), and a SoftMax layer are all included in the convolutional block γ. The global average pooling layer sends the average values of each element on the feature map to the following layer. SoftMax gives each output categorization result in a probability value:

$$\tilde{z}_i \triangleq softmax(z_i) = \frac{e^{z_i}}{\sum_{\forall} e^{z_i}}, \tag{2}$$

where z_i is the value of different nodes. Two CNN paths to learn information in different frequency bands.

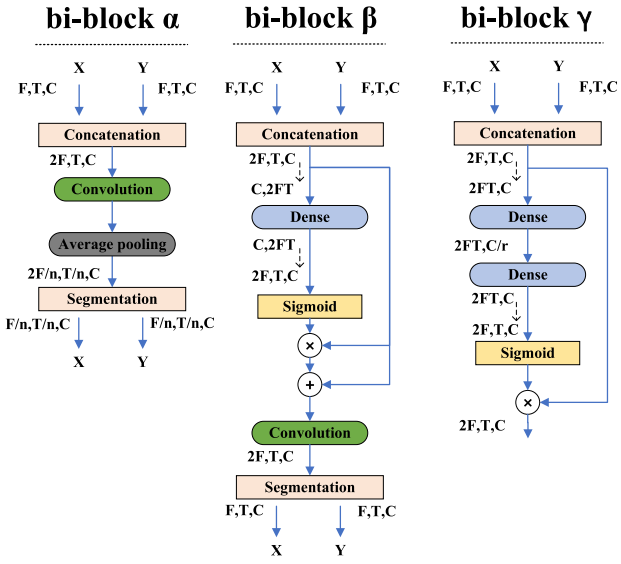

Fig. 2 Illustration of the bi-blocks.

2.3 Bi-Blocks

Three band interaction blocks are used in the designed model: bi-block α, bi-block β, and bi-block γ, as shown in Fig. 2.

First, bi-block α helps the two CNN paths obtain more receptive fields, especially the additional time-frequency information at the frequency band connection. The inputs of bi-block α are high-frequency component $f^{high} \in \mathbb{R}^{F\times T\times C}$ and low-frequency component $f^{low} \in \mathbb{R}^{F\times T\times C}$, where F, T, and C are the dimensions of frequency, time, and number of channels, respectively. These frequency band components are first spliced on the frequency dimension, and the input feature can be expressed as:

$$f \triangleq [f^{high}, f^{low}] \in \mathbb{R}^{2F\times T\times C}. \tag{3}$$

Then, a convolution layer with a larger convolution kernel is used to extract the time-frequency correlation infor-

mation of the global feature, and an average pooling layer is used to match the output dimensions. The feature is then separated into different frequency bands, and the obtained information is appended to each path. Note that the fusion of the output of the features from bi-block α and the original paths uses channel merging instead of element addition.

The bi-block β captures non-local information between two paths. First, create a global feature, the same as bi-block α, by fusing the frequency axis of high-frequency and low-frequency components. Then, the input feature's time dimension and feature dimension are combined to form a two-dimensional tensor. These tensors gather non-local information from the dense layer and use a Sigmoid layer to create an attention feature map:

$$y_\beta(f_i) = \sigma(f_i W_y), \tag{4}$$

where $W_y \in \mathbb{R}^{2FT\times 2FT}$ is a weight vector of dense layer, σ is the activation function sigmoid. In addition, '$\otimes$' represents the element-wise Hadamard product, '$\oplus$' denotes the element-wise sum in Fig. 2. After the self-attention feature is obtained, the convolution layer is then utilized to get additional time-frequency information, and the frequency band is then once again split.

The main function of bi-block γ is channel attention calculation, which is similar to CBAM [12]. First, $f \in \mathbb{R}^{2FT\times C}$ as input, the same as bi-block β. Then, the feature is sent to the two-layer dense to generate the final channel attention feature map:

$$y_\gamma(f_i) = \sigma(f_i W_\theta W_\phi), \tag{5}$$

here $W_\theta \in \mathbb{R}^{C\times\frac{C}{r}}$, $W_\phi \in \mathbb{R}^{\frac{C}{r}\times C}$, r is the dimension reduction parameter. Then, a Sigmoid layer obtains the feature map based on channel attention. In addition, the frequency band of the input feature is no longer divided, and the global feature is used as the module's output.

3. Experiments

3.1 Datasets and Training Setup

We evaluate our proposed DCNN-bi on the open dataset in the ASC task of DCASE 2018 task1A and DCASE 2020 task1A [13]. Ten scene environments from six major European cities are recorded in the DCASE database. Both of them contain 10-second-long audio clips.

The audio clips of DCASE 2018 are first resampled to 48 kHz, and the audio clips of DCASE 2020 are resampled to 44.1 kHz. Then, using a Hamming window with a length of 2048 samples, an overlap of 1024, and 128 mel bins, the log mel-spectrograms are extracted from the audio clips. In addition, the first and second derivatives of the spectrogram are calculated to form a three-dimensional feature as the input of the model. For DCASE 2018, enter the feature dimension as 461×128×3, and 423×128×3 for DCASE 2020. Since we need to divide the mel-spectrograms into high-frequency bands and low-frequency bands to meet the needs of CNN, we divide the first 64 frequency points into low-frequency bands and the last 64 frequency points into high-frequency bands.

Table 1 Our baseline DCNN-bi model for ASC.

Modules	layer	Shapes
	Convolution	3×3, stride [2,1] kernels 48
Convblock α	Convolution	3×3, stride [1,1] kernels [48, 96, 192]
	Convolution	3×3, stride [1,1] kernels [48, 96, 192]
	Average pooling	3×3, stride [2,2]
bi-block α	Convolution	7×7, stride [1,1] kernels [48, 96, 192]
	Average pooling	3×3, stride [2,2]
Convblock β	Convolution	3×3, stride [1,1] kernels [384, 768]
	Convolution	3×3, stride [1,1] kernels [384, 768]
bi-block β	Dense	$T\times F$
	Convolution	3×3, stride [1,1] kernels [384, 768]
	Dense	384, $r = 4$
bi-block γ	Dense	1536
Convblock γ	Convolution	1×1, stride [1,1] kernels 1536
	Convolution	1×1, stride [1,1] kernels 10
	GAP	
	SoftMax	

Table 1 shows the proposed model baseline. Where $T\times F$ represents the shape of the frequency dimension of the input tensor multiplied by the time dimension. r is the parameter reduction coefficient. In the process of model training, we use a stochastic gradient descent optimizer with a batch size of 64, momentum of 0.9, the learning rate is initialized to 0.01, and a cosine-decay-restart learning rate scheduler is used to reset the learning rate to 0.01 after 3, 7, 15, 31 and 64 epochs. In addition, we used Mixup and spectrum augment in training, and we implemented DCNN-bi in TensorFlow.

3.2 Experiment Results

We first analyze the influence of the three bi-blocks on the performance of ASC. The basic system is the DCNN model without any additional bi-blocks. Table 2 displays a performance comparison of all experimental models. The findings demonstrate that the performance of DCNN is significantly improved with the increase of bi-blocks, and the recognition rates of the two acoustic scene data sets are raised by 1.79% and 3.06%, respectively. According to the experimental findings, the bi-blocks can assist the DCNN framework in achieving more accuracy through information interaction and late fusion of various frequency paths.

Subsequently, we proceed to examine the enhancements in performance exhibited by the bi module when subjected to various baseline CNN architectures. Table 3 presents the results of comparative experiments, wherein the number of convolutional kernels in DCNN_1 is halved in comparison to

Table 4 ASC performance (%) for each scene.

Dataset	Method	airport	bus	metro	metro station	park	public square	shopping mall	street pedestrian	street traffic	tram	Avg
DCASE 2018	DCNN	75.09	80.58	69.73	92.28	91.74	45.83	81.00	65.59	91.87	80.84	77.45
	DCNN w/(α)	69.81	85.95	73.95	91.51	91.32	48.15	85.66	70.85	92.28	76.63	78.61
	DCNN w/(α,β)	71.7	74.79	75.86	94.98	88.43	51.85	83.51	64.37	90.65	82.38	77.85
	DCNN w/(α,β,γ)	72.08	79.75	80.84	94.21	90.91	48.61	83.15	70.04	92.68	80.08	79.23
DCASE 2020	DCNN	55.41	82.49	62.63	71.72	89.9	52.19	65.66	47.14	86.53	71.62	68.52
	DCNN w/(α)	54.73	82.83	68.01	75.42	90.91	43.1	70.71	57.91	88.89	78.38	71.08
	DCNN w/(α,β)	55.74	81.14	65.99	74.75	87.21	53.87	68.35	57.24	87.21	79.39	71.08
	DCNN w/(α,β,γ)	71.62	82.49	72.73	70.03	85.19	56.9	61.95	49.83	87.21	78.04	71.59

Table 2 Comparison of the recognition rate (%) of different bi-blocks.

Network	bi-block	2018	2020
DCNN	/	77.87	68.53
DCNN	α	78.20	71.05
DCNN	α,β	78.27	71.09
DCNN	α,β,γ	79.66	71.59

Table 3 Comparison of the recognition rate (%) of different DCNN.

Network	2018	2020	complexity
DCNN	77.87	68.53	22M
DCNN-bi	79.66	71.59	41M
DCNN_1	73.47	61.62	1.4M
DCNN_2	74.14	63.14	3.1M
DCNN-bi_1	74.50	65.56	3M

Table 5 Comparison of the recognition rate (%) of different methods.

Methods	DCASE 2018	DCASE2020
DCASE baseline [13]	59.7	51.6
MCTA-CNN [14]	72.40	/
SeNoT-Net [15]	77.19	/
Xception [16]	79.8	/
Trident ResNet [17]	/	73.7
1aCNN [18]	/	62.1
TRN Dev [19]	/	70.3
DCNN-bi	79.66	71.59

DCNN, and DCNN_2 is further adjusted to achieve a parameter quantity that is approximately equivalent to DCNN-bi_1. In other words, it is possible to evaluate and compare the performance of various models by considering the same parameter quantity. The analysis of Table 3 reveals that the incorporation of bi-modules in low-complexity CNN models leads to an enhancement in the performance of the baseline CNN. Moreover, even when considering the same parameter quantity, the DCNN-bi model exhibits enhanced performance compared to the baseline CNN model without relying solely on increasing the number of parameters.

Table 4 shows the category precision of baseline and DCNN-bi on two data sets. We can infer that DCNN-bi has a great improvement in the class statistics, especially for metro, public square, street pedestrian, and street traffic.

Additionally, as indicated in Table 5, we contrast the approach we developed with some CNN-based works. The comparative experiment mainly consists of two parts: [14] and [15] are CNN-based ASC systems, [16]–[19] are improved models based on DCASE task1a baseline CNN. In addition, [13] is the baseline CNN model for DCASE task1a.

The table illustrates that the performance of the proposed DCNN-bi surpasses the official dataset baseline by 19.96% and 19.99% in two datasets, respectively. This improvement is superior to the majority of comparative ASC models. Nevertheless, we observed performance is comparatively inferior to that reported in [16] and [17], potentially attributable to the outdated architectural design of the baseline CNN, which encourages us to further enhance the pertinent ASC models in our forthcoming research endeavors.

4. Conclusion

This paper presented a dual-path convolutional neural network based on a band interaction block for ASC, which has two paths to learn the high-frequency and low-frequency components of the input spectrum. Additionally, three bi-blocks are further arranged in the two paths, bi-block α helps the two DCNN paths obtain more receptive fields, bi-block β is used to capture non-local information, and bi-block γ focuses on channel attention calculation. The outcomes of the experiments demonstrate the effectiveness of the DCNN-bi in enhancing system performance.

Acknowledgments

This research was supported by the National Key Research and Development Program of China under Grant No. 2020YFC2004003, the National Natural Science Foundation of China under Grant No. 62001215.

References

[1] J. Abeßer, "A review of deep learning based methods for acoustic scene classification," Applied Sciences, vol.10, no.6, p.2020, 2020.
[2] D. Barchiesi, D. Giannoulis, D. Stowell, and M.D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," IEEE Signal Process. Mag., vol.32, no.3, pp.16–34, 2015.
[3] H. Hu, C.H.H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S.M. Siniscalchi, Y. Wang, J. Du, and C.H. Lee, "A two-stage approach to device-robust acoustic scene classification," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.845–849, 2021.
[4] J.H.K. Soonshin Seo, "Mobilenet using coordinate attention and fusions for low-complexity acoustic scene classification with multiple devices," Technical Report, DCASE2021 Challenge, 2021.
[5] W. Gao and M. McDonnell, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," Technical Report, DCASE2019 Challenge, June

2019.

[6] H. Liang and Y. Ma, "Acoustic scene classification using attention-based convolutional neural network," Technical Report, DCASE2019 Challenge, June 2019.

[7] J.-W. Jung, H.-S. Heo, H.-J. Shim, and H.-J. Yu, "DNN based multi-level features ensemble for acoustic scene classification," Technical Report, DCASE2018 Challenge, Sept. 2018.

[8] M.M. Morgan, I. Bhattacharya, R.J. Radke, and J. Braasch, "Classifying the emotional speech content of participants in group meetings using convolutional long short-term memory network," J. Acoust. Soc. Am., vol.149, no.2, pp.885–894, 2021.

[9] L. Jie, "Acoustic scene classification with residual networks and attention mechanism," Technical Report, DCASE2020 Challenge, June 2020.

[10] Y. Lee, S. Lim, and I.Y. Kwak, "The CAU-ET acoustic scenery classification system for DCASE 2020 challenge," Technical Report, DCASE2020 Challenge, June 2020.

[11] L. Mingle and L. Yanxiong, "The system for acoustic scene classification using resnet," Technical Report, DCASE2019 Challenge, June 2019.

[12] S. Woo, J. Park, J.Y. Lee, and I.S. Kweon, "CBAM: Convolutional block attention module," Proc. European Conference on Computer Vision (ECCV), Munich, Germany, pp.3–19, 2018.

[13] T. Heittola, A. Mesaros, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," Technical Report, DCASE2018 Challenge, Sept. 2018.

[14] Y. Wang, C. Feng, and D.V. Anderson, "A multi-channel temporal attention convolutional neural network model for environmental sound classification," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.930–934, 2021.

[15] L. Zhang, J. Han, and Z. Shi, "Learning temporal relations from semantic neighbors for acoustic scene classification," IEEE Signal Process. Lett., vol.27, pp.950–954, 2020.

[16] Y. Liping, C. Xinxing, and T. Lianjie, "Acoustic scene classification using multi-scale features," Technical Report, DCASE2018 Challenge, Sept. 2018.

[17] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," Technical Report, DCASE2020 Challenge, June 2020.

[18] R. Abbasi and P. Balazs, "Acoustic scene classification by the snapshot ensemble of CNNs with XGBoost," Technical Report, DCASE2020 Challenge, June 2020.

[19] K. Vilouras, "Acoustic scene classification using fully convolutional neural networks and per-channel energy normalization," Technical Report, DCASE2020 Challenge, June 2020.