2020 RISP International Workshop on Nonlinear Circuits,
Communications and Signal Processing (NCSP 2020)
Honolulu, Hawaii, USA, February 28 - March 2, 2020

Spatial Feature Extraction Based on Convolutional Neural Network with Multiple Microphone Inputs for Monitoring of Domestic Activities

Yuki Kaneko, Rika Kurosawa, Takeshi Yamada, and Shoji Makino

University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan
E-mail: y.kaneko@mmlab.cs.tsukuba.ac.jp

Abstract

In acoustic scene classification, the use of multiple microphone inputs enables to improve classification performance by utilizing spatial information. Recently, a method of spatial feature extraction using CNNs (convolutional neural networks), 2D-CNN, has been proposed. In this method, CNNs are separately applied to the time-frequency domain, time-space domain, and frequency-space domain, respectively. Furthermore, a method using CNN with a three-dimensional filter, 3D-CNN, has been proposed. It extracts spatial features across the time-frequency-space domain. Both methods intend to extract spatial features suitable for classification. In this paper, to examine an appropriate structure of CNN for extracting spatial features, we compared 2D-CNN and 3D-CNN by applying them to the dataset with four microphone inputs prepared for DCASE2018 Task 5. The experimental results confirmed that the F-score of 3D-CNN is better than that of 2D-CNN for different number of microphone inputs. The best F-score of 87.7 was given by 3D-CNN with three microphone inputs.

1. Introduction

In recent years, interest in improvements to the quality of life such as smart homes has been increasing. Typical functions of a smart home include a system for monitoring children and elderly people and smart speakers that turn on the light and play music in response to the vocal instructions of people. In a smart home, the information acquired using various sensors is utilized, and sound information also plays an important role.

Extracting sound information from the sound generated in a certain environment is called environmental sound recognition. The DCASE (Detection and Classification of Acoustic Scenes and Events) Challenge [1] has been held continuously since 2013 for the purpose of improving the technical level of environmental sound recognition. As DCASE2018 Task 5, an acoustic scene classification task was provided to classify the sounds recorded at home into nine classes, such as cooking and television. Since the dataset in this task was recorded using four channels (microphones), the performance of acoustic scene classification can be expected to be improved by utilizing spatial information. However, many methods submitted for this task utilized each individual signal recorded using multiple channels, so did not make full use of spatial information. The method with the best performance among the methods submitted for this task focused on data augmentation and did not use spatial features explicitly[2]. It is therefore necessary to consider how to utilize spatial information contained in multichannel signals.

Recently, a method of spatial feature extraction using CNNs (convolutional neural networks) with a two-dimensional filter, 2D-CNN, has been proposed[3]. This method extracts spatial features suitable for classification by separately applying CNNs to the time–frequency domain, time–space domain and frequency–space domain. The effectiveness of this method was shown in experiments using two channel signals. Furthermore, a method using CNN with a three-dimensional filter, 3D-CNN, has been proposed [4]. This method extracts features across the time–frequency–space domain by applying CNN with a three-dimensional filter. In this paper, to examine a suitable structure of CNN for extracting spatial features, we apply 2D-CNN and 3D-CNN to DCASE2018 Task 5 and compare their performance. We also investigate the effect of increasing the input channels.

2. Spatial feature extraction using CNNs

Fig. 1 shows the process flow of the method of spatial feature extraction using CNNs[3]. In this method, the input signal is assumed to be a 10 s stereo signal, and a 128-dimensional log Mel filter bank output is calculated for both the left and right channels. CNN is then applied to each domain. The blue squares in Fig. 1 represent the two-dimensional filter for each domain, and the upper, middle, and bottom squares represent the CNNs being applied to the time–frequency domain, time–space domain and frequency–space domain, respectively. For example, in the CNN for the time–frequency domain, the time–frequency spectrogram is

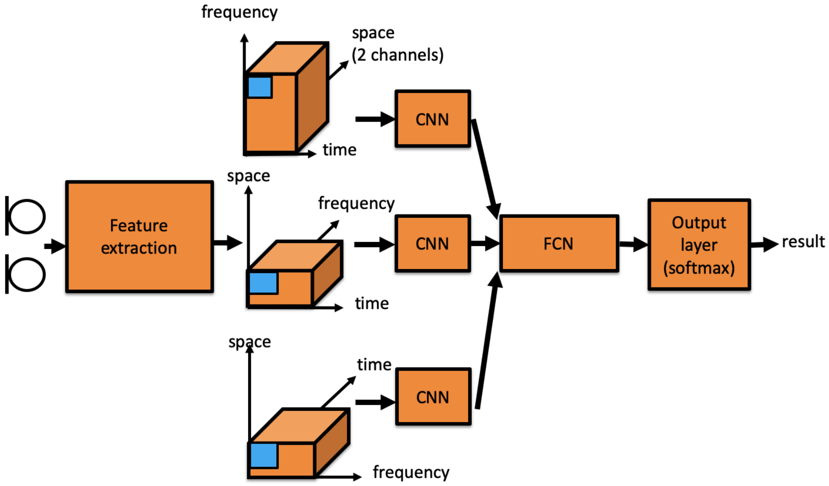
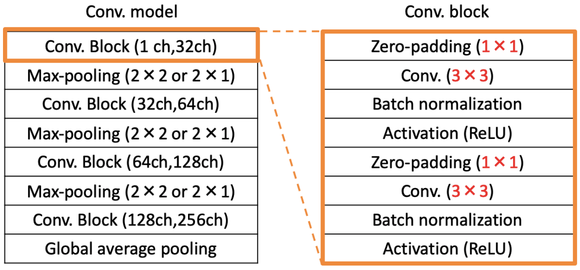


Figure 1: Process flow of spatial feature extraction method using CNNs



Conv. model
Conv. Block (1 ch,32ch)
Max-pooling (2 × 2 or 2 × 1)
Conv. Block (32ch,64ch)
Max-pooling (2 × 2 or 2 × 1)
Conv. Block (64ch,128ch)
Max-pooling (2 × 2 or 2 × 1)
Conv. Block (128ch,256ch)
Global average pooling

Conv. block
Zero-padding (1 × 1)
Conv. (3 × 3)
Batch normalization
Activation (ReLU)
Zero-padding (1 × 1)
Conv. (3 × 3)
Batch normalization
Activation (ReLU)

Figure 2: Conv. model and conv. block in the 2D-CNN

regarded as a feature map and space (stereo) is regarded as the channels of the CNN. Therefore, the number of channels of the CNN is two. On the other hand, in the CNN for the time–space domain, since frequency is regarded as the channels of the CNN, the number of channels of the CNN is 128, the number of Mel frequency bins. Convolutions for the time–space domain and frequency–space domain correspond to extracting spatial information between stereo channels in a feature map.

In this method, the CNN consists of eight convolution layers. Fig. 2 shows the conv. model and conv. block when the number of channels of each CNN is one. In the conv. model, the conv. block that performs convolutions is repeatedly performed with max-pooling. The two numbers separated by a comma in the parentheses of the conv. model are the number of input channels in the conv. model on the left and the number of output channels on the right. The numbers in the parentheses of the pooling layer are the window sizes used for pooling. The window size in the pooling layer is 2×2 for the time–frequency domain. On the other hand, for the time–space domain and frequency–space domain, the window size in the pooling layer is 2×1. This is because it maintains the dimension of the space. The total dimension of the feature map after global average pooling is $256 \times n$, where n is the number of channels inputted to each CNN.

Finally, we explain the conv. block. In the conv. block, zero padding with a size of 1×1 is first performed for the input feature map. Next, a convolution with filters of kernel size 3×3 is performed. Then, batch normalization [5] is conducted. Finally, the ReLU function is applied as an activation function. These processes are performed twice in the conv. block.

The effectiveness of this method was confirmed by an experiment using stereo signals. In this paper, we apply this method to the dataset with four channel signals prepared for DCASE2018 Task 5. The part shown in red in Fig. 2 is the part whose size has been changed. The kernel size of the convolution is the same as that in the above method in the time–frequency domain. On the other hand, the kernel size of the convolution is changed to 3×5 in the time–space domain and frequency–space domain. This means that the dimension of the space is 5. This change enables to extract spatial features spanning two or more microphones. The convolution is performed while maintaining the number of channels in the same way as in the above method. The zero padding in the time–frequency domain is 1×1. On the other hand, the zero padding in the time–space domain and the frequency–space domain is 1×2. This is to prevent the reduction of the dimension when performing convolution.

3. Spatial feature extraction with a three-dimensional filter

The process flow of 3D-CNN supposing two channels as the input is shown in Fig. 3. The blue box represents a three-

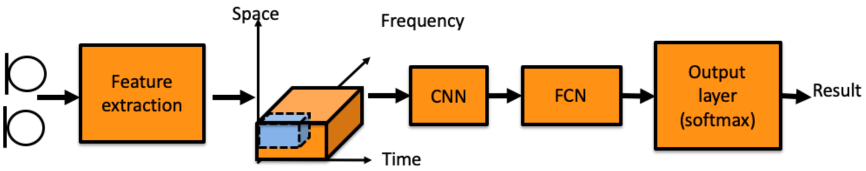
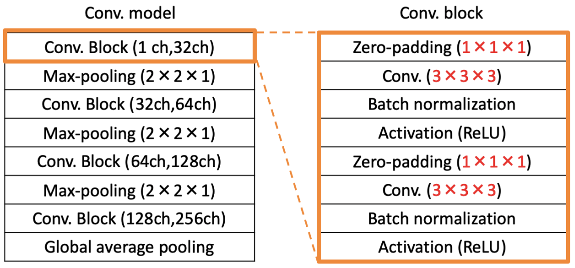


Figure 3: Process flow of the 3D-CNN



Conv. model
Conv. Block (1 ch,32ch)
Max-pooling (2 × 2 × 1)
Conv. Block (32ch,64ch)
Max-pooling (2 × 2 × 1)
Conv. Block (64ch,128ch)
Max-pooling (2 × 2 × 1)
Conv. Block (128ch,256ch)
Global average pooling

Conv. block
Zero-padding (1 × 1 × 1)
Conv. (3 × 3 × 3)
Batch normalization
Activation (ReLU)
Zero-padding (1 × 1 × 1)
Conv. (3 × 3 × 3)
Batch normalization
Activation (ReLU)

Figure 4: Conv. model and conv. block in the 3D-CNN

Table 1: Overview of the dataset

# of scenes	9
# of signals in development set	72984
# of signals in evaluation set	72972
# of channels	4
microphone interval	5 cm
sampling frequency	16 kHz
quantization bits	12

dimensional filter when applying the CNN. This method can extract features across the time–frequency–space domain and the number of parameters in the CNN is significantly reduced as compared with the 2D-CNN. Fig. 4 represents the conv. model and conv. block when the number of channels of the CNN is one. These are basically the same as those in Fig. 2. The numbers of input and output channels, the kernel size of the convolution, and the window size of pooling are designed in accordance with the same objective as in Fig. 2.

When this method is applied with four channels as the input, the kernel size of the convolution is changed. The kernel size of the convolution is $3 \times 3 \times 5$ and the size of zero padding is $1 \times 1 \times 2$. The dimensions of the space are 5 and 2, respectively. The reason for this size is the same as that for the two-dimensional filter.

4. Experiment

We use the dataset provided in DCASE2018 Task 5 for the experiment. This dataset is part of the SINS dataset [6]. This dataset comprises the sounds of a person living in a villa recorded using four-channel microphone arrays for a week. The arrangement of the microphone arrays in the room is shown in Fig. 5. Table 1 shows an overview of the dataset. There are nine types of sound scene, and the total number of signals in the development dataset is 72984. The number of signals of each sound scene is shown in Table 2. " # sessions" indicates the number of recordings of the sound signals of each class, and each sound signal is divided into segments of 10 s, the number of which is given by "# 10 s segments." The number of sound signals for dishwashing and vacuum cleaning is small while the number of sound signals for absence, working and watching TV is large. The number of datasets for evaluation is 72972, the microphone interval in the microphone array is 5 cm, and the sampling frequency is 16 kHz. The quantization bits is 12. We train the classifier using the development dataset and evaluate it using the evaluation dataset. To evaluate only the effect of spatial feature extraction, we do not use external data or data augmentation.

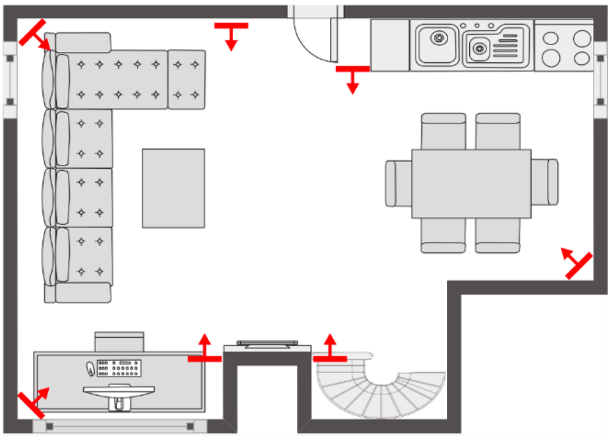


Figure 5: Layout of the microphone arrays (quoted from [1])

Table 3 shows the conditions of acoustic features and CNNs[3]. The features are 128th-order log mel filter bank outputs. The frame length and frame shift length in the frame analysis are 40 and 20 ms, respectively. When we apply CNNs to features, we divide the time series of the features

Table 2: Number of acoustic signals of each class in the development set

activity	# sessions	# 10 s segments
absence	42	18860
cooking	13	5124
dishwashing	10	1424
eating	13	2308
working	33	18644
social activity	21	4944
vacuum cleaning	9	972
watching TV	9	18648
other	118	2060
total	268	72984

Table 3: Conditions of the acoustic features and CNNs

acoustic feature	128th-order log mel filter bank outputs
frame length	40 ms
frame shift length	20 ms
audio block size	10 frame (without overlap)
# of conv. layers	8
optimization method	Adam
epoch	50

into blocks with ten frames. Adam [7] is used as the optimization method for training the classifier, and the number of epochs during training is set up to 50 and the epoch that gives the best F-score is adopted. Chainer [8] is used for the implementation of the 2D-CNN and 3D-CNN.

Table 4 shows the F-scores of 2D-CNN and 3D-CNN for different number of input channels. The F-scores of 3D-CNN are better than that of 2D-CNN in any channels. It shows that features across the time–frequency–space domain give better results. Also, we examine the effectiveness of increasing the number of input channels. In both methods, increasing the number of input channels improves the F-score. This is because more rich spatial information becomes available by using three or more microphone inputs. The best F-score was given by 3D-CNN with three channels, which is 87.7%. This is equivalent to the 5th place in DCASE2018 Task 5. The use of data augmentation technique, like the top ranking methods, would give better results.

5. Conclusion

In this paper, we compared 2D-CNN and 3D-CNN to examine a suitable structure of CNN for extracting spatial features. The experimental results confirmed that the classification performance is improved by expanding the filter to three dimensions and further increasing the number of input channels. The best F-score of 87.7% was given by 3D-CNN with three channels.

Table 4: F-score of each method for different number of input channels

	2ch	3ch	4ch
2D-CNN	84.6	86.5	85.2
3D-CNN	85.4	87.7	86.8

Acknowledgment

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant No. 19H04131.

References

[1] DCASE 2018 Challenge Task 5 website, http://dcase.community/challenge2018/task-monitoring-domestic-activities.

[2] T. Inoue, P. Vinayavekkhin, S. Wang, D. Wood, N. Greco, R. Tachibana, “ Domestic activities classification based on CNN using shuffling and mixing data augmentation, ” DCASE2018 Challenge Technical Report, 2018.

[3] G. Takahashi, T. Yamada, S. Makino, “ Acoustic scene classification based on spatial feature extraction using convolutional neural networks, ” Proc. NCSP2018, pp. 351-354, Mar. 2018.

[4] S. Adavanne, A. Politis, T. Virtanen, ”Multichannel sound event detection using 3D convolutional neural networks for learning inter-channel features,” Proc. IJCNN2018, pp. 1–7, July 2018.

[5] S. Ioffe, C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” ArXiv:1502.03167v1, Feb. 2015.

[6] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. V. D. Bergh, T. V. Waterschoot, B. Vanrumste, M. Verhelst, P. Karsmakers, “ The sins database for detection of daily activities in a home environment using an acoustic sensor network, ” Proc. DCASE2017, pp. 32-56, Nov. 2017.

[7] D. P. Kingma, J. L. Ba, “Adam: A method for stochastic optimization,” ArXiv:1412.6980v1, Dec. 2014.

[8] Chainer website, https://chainer.org/.