LETTER

A CNN-Based Feature Pyramid Segmentation Strategy for Acoustic Scene Classification

Ji XI[†a)], ***Nonmember***, **Yue XIE**[††], ***Member***, **Pengxu JIANG**[†††], ***and*** **Wei JIANG**[†], ***Nonmembers***

SUMMARY Currently, a significant portion of acoustic scene categorization (ASC) research is centered around utilizing Convolutional Neural Network (CNN) models. This preference is primarily due to CNN's ability to effectively extract time-frequency information from audio recordings of scenes by employing spectrum data as input. The expression of many dimensions can be achieved by utilizing 2D spectrum characteristics. Nevertheless, the diverse interpretations of the same object's existence in different positions on the spectrum map can be attributed to the discrepancies between spectrum properties and picture qualities. The lack of distinction between different aspects of input information in ASC-based CNN networks may result in a decline in system performance. Considering this, a feature pyramid segmentation (FPS) approach based on CNN is proposed. The proposed approach involves utilizing spectrum features as the input for the model. These features are split based on a preset scale, and each segment-level feature is then fed into the CNN network for learning. The SoftMax classifier will receive the output of all feature scales, and these high-level features will be fused and fed to it to categorize different scenarios. The experiment provides evidence to support the efficacy of the FPS strategy and its potential to enhance the performance of the ASC system.

key words: *spectrum features, convolutional neural network, feature pyramid segmentation, deep learning*

1. Introduction

The primary objective of acoustic scene classification (ASC) is to categorize the audio input of a given model into predefined scenes, and ASC-based systems have a wide range of applications. Currently, researchers primarily focus on the detection and classification of acoustic scenes and events (DCASE) within the context of ASC, and the annual ASC-based challenge and public dataset in DCASE have played a significant role in fostering the advancement of ASC. Most scholars in the field of ASC predominantly employ neural network techniques to categorize auditory scenes. In contrast to early machine learning techniques, deep learning approaches exhibit superior capability in extracting pertinent scene information from audio data.

Presently, neural network models based on ASC predominantly depend on using CNN. [1] propose an efficient genetic algorithm (GA) that aims to find optimized CNN architectures for the ASC task. [2] propose to employ residual quaternion CNNs for low complexity, device-robust ASC. The proposed model RQNet uses quaternion encoding to increase the accuracy with fewer parameters. [3] propose the R-Block, to explore the relation information in an explicit and comprehensive way. Furthermore, it is worth noting that most of the prominent models utilized in the DCASE challenge are founded upon CNN. CNN-based ASC models frequently utilize a two-dimensional spectrum as the input to the model. The CNN can effectively capture the temporal and frequency features in the spectrum due to the similarity of its spectrum properties to those of image representations. The proposition regarding relevant CNN networks reinforces this network's prevailing influence inside the realm of ASC.

Nevertheless, owing to the unique properties of spectrum features, distinctions exist in the attributes of spectrum and pictures. The target object's location in the image is not constrained; nevertheless, it is essential to note that objects appearing in different frequency bands in the spectrum can possess distinct physical interpretations. Therefore, it is crucial to divide and analyze the frequency component of the input spectrum. Moreover, it is crucial to carefully examine the frequency aspect and conduct a segmented study of the temporal dimension. This is because scene audio data typically exhibits both periodic and random characteristics.

This article proposes a CNN-based network for ASC based on the FPS strategy. The proposed system flowchart is shown in Fig. 1. Like most CNN-based ASC networks, our developed CNN utilizes spectrum as input features. Building upon the concept of spatial pyramid pooling (SPP) [4], we used feature pyramid segmentation (FPS) on the input spectrum to obtain the spectrum information over various time and frequency bands. Specifically, FPS is applied to input spectrogram features to obtain multiple input features of the same scale. Subsequently, these feature maps of different scales are used as inputs for different CNNs. In addition, parallel CNN modules have the same structural parameters. Finally, these feature maps of the same scale are combined

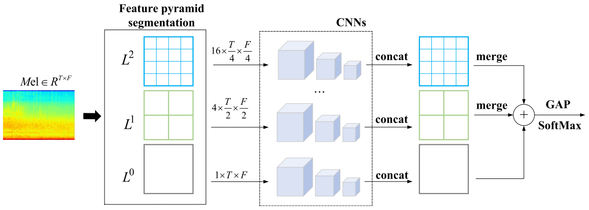
Feature pyramid segmentation
CNNs
$Mel \in R^{T\times F}$
L^2
L^1
L^0
$16\times\frac{T}{4}\times\frac{F}{4}$
$4\times\frac{T}{2}\times\frac{F}{2}$
$1\times T\times F$
...
concat
concat
concat
merge
merge
GAP
SoftMax

Fig. 1 Illustration of the proposed model.

Manuscript received February 20, 2024.
Manuscript publicized March 26, 2024.

[†] School of Computer Information Engineering, Changzhou Institute of Technology, P.R. China.
[††] School of Communication Engineering, Nanjing Institute of Technology, P.R. China.
[†††] School of Information Science and Engineering, Southeast University, P.R. China.
a) E-mail: xiji@czust.edu.cn

DOI: 10.1587/transinf.2024EDL8013

into a global feature representation, and the outputs of multiple CNNs are fused as the outputs of the designed model.

2. System Description

2.1 Input Feature

Spectrum is extensively utilized in the fields of speech detection and audio processing. Currently, the primary input features utilized in ASC-based CNN consist of spectrum features. As a result, we use spectrum as input features in our model. The mel-spectrogram, which falls under the spectrum feature category, can extract frequency information from speech or audio data. The primary stages of extracting the Mel spectrum encompass pre-weighting, framing, windowing, and applying a fast fourier transform on the original speech signal. This process facilitates the conversion of the time-domain signal into a frequency-domain signal. Subsequently, it is essential to analyze the signal in the frequency domain by applying a sequence of Mel filters, thereby obtaining spectrum attributes that are derived from the Mel spectrum. The spectrum features that have been retrieved consist of two-dimensional matrices. These matrices have two dimensions, namely time and frequency. Each element inside these matrices reflects the logarithmic amplitude value of the relevant frequency channel. This particular feature representation enables a more effective capture of the frequency information inherent in audio signals.

2.2 Convolutional Neural Network

CNN is a prevalent deep learning model extensively employed in computer vision, particularly for tasks related to image recognition. CNN is extensively utilized in several domains owing to its exceptional capability for feature extraction and the advantageous utilization of parameter sharing. Using two-dimensional spectrograms as input to the model is common in ASC, making CNN a prevalent choice in the field.

CNN typically includes an input layer, convolutional layer, activation layer, pooling layer, fully connected layer, global pooling layer, and output layer.

Input layer: The input of CNN is two-dimensional data, usually images or other forms of two-dimensional data.

Convolutional layer: The convolutional layer is the core part of CNN. It extracts local features of the image by performing convolution operations on the input image with a series of convolution kernels. Each convolution kernel moves on the input image through a sliding window, calculating the convolution of the image regions within the window one by one and generating corresponding feature maps. The convolutional layers can be represented as:

$$y_{i,j,k} = f\left(\sum_{a=0}^{h-1}\sum_{b=0}^{w-1}\sum_{c=0}^{c'-1} x_{i+a,j+b,c} \times k_{a,b,c,c'} + b_k\right), \tag{1}$$

here, x represents the input feature map, k is the convolution kernel, b_k is the bias term, h and w are the height and width of the kernel, f is the activation function, and y is the output feature map of the convolutional layer. The weights of convolutional kernels are automatically learned during the training process to capture different features.

Activation function: After the convolutional layer, a nonlinear activation function, such as ReLU, is usually added, which turns all negative numbers to zero and retains positive numbers. ReLu can be represented as:

$$y = \max(0, x), \tag{2}$$

here, x is the input, and y is the output of the ReLU activation layer. The activation function introduces nonlinearity, increasing the network's expressive power.

Pooling layer: The pooling layer is used to reduce the spatial size and number of parameters of feature maps. Maximum pooling and average pooling are commonly used pooling operations. They reduce the representation size by taking the maximum or average value within each window. The pooling operation helps to extract main features and maintain translation invariance.

Fully connected layers: After passing through multiple convolutional and pooling layers, one to more fully connected layers are usually added. Each neuron in the fully connected layer is connected to all neurons in the previous layer. The fully connected layer performs classification or regression tasks by learning higher-level features. The fully connected layer can be represented as:

$$y_j = f\left(\sum_{i=1}^{n} w_{i,j}x_i + b_j\right), \tag{3}$$

here, x is the input vector, w is the weight matrix, b is the bias vector, f is the activation function, and y is the output vector of the fully connected layer.

Global pooling layer: The Global pooling layer (GAP) is a particular pooling layer commonly used in CNN. Unlike traditional local pooling layers, the global pooling layer performs pooling operations on the entire feature map, summarizing the information of all feature positions to generate a fixed-size output. The GAP layer can be represented as:

$$y_k = \frac{1}{h \times w}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1} x_{i,j,k}, \tag{4}$$

here, x represents the input feature map, h and w are the height and width of the pooling window, and y is the output feature map of the global pooling layer.

Output layer: The output layer is determined based on the requirements of specific tasks. For classification tasks, SoftMax is usually used as the activation function of the output layer to generate the category probability distribution. Linear activation functions or other appropriate functions can be used for regression tasks.

In addition, CNN has advantages in local feature extraction, translation invariance, parameter sharing, automatic learning feature representation, and scalability in classification tasks. This enables CNN to perform ASC tasks effectively.

2.3 Feature Pyramid Segmentation Strategy

The feature pyramid segmentation technique was derived from the spatial pooling strategy and implemented on the input spectrogram to capture localized information across several temporal and frequency domains. The depicted technique for FPS is illustrated in Fig. 1. Specifically, different segmentation strategies exhibit variations in terms of feature scales. The segmentation technique of the feature pyramid involves partitioning the input feature map into grid areas of varying sizes, with the selection of grid region sizes being dependent on the size of the input feature map. As illustrated in Fig. 1, the segmentation approach denoted as $L2 = [L^0, L^1, L^2]$ encompasses three distinct segmentation methods. These methods enable the division of the input feature map into grid sections of varying dimensions, namely 1×1, 2×2, and 4×4. This entails partitioning the input features into consecutive local features of one, four, and sixteen segments. Subsequently, the segment-level features will be directed towards distinct CNN pathways to acquire time-frequency information at varying scales.

3. Experiments

3.1 Experimental Setup

The DCASE 2018, DCASE 2019, and DCASE 2021 [5] datasets were used as the evaluation datasets for the model. DCASE is an international competition for the Detection and Classification of Acoustic Scenes and Events. DCASE aims to promote research in environmental sound and provide standard datasets to help researchers compare the performance of different algorithms. All datasets have the same ten different recording environments, each lasting approximately 10 seconds. The DCASE 2018 has 6122 files for training and 2518 for testing; the DCASE 2019 has 9185 files for training and 4185 for testing; the DCASE 2021 has 13962 files for training and 2968 for testing.

The mel-spectrogram is the input feature of our model. For each audio data, 128 mel filter banks were used to obtain mel spectral features, using a frame size of 2048 samples and a Hamming window of 1024 hops. The sampling frequency is set to 48 kHz for DCASE2018 and DCASE2019, 44.1 kHz for DCASE2021. In addition, the detailed parameters of the designed CNN model are shown in Table 1, where the convolutional layer is connected to the activation layer and batch normalization layer.

For the training phase of the model, we use a stochastic gradient descent optimizer with a batch size of 64, momentum of 0.9, and the learning rate is initialized to 0.01. In addition, we used Mixup and spectrum augment in training.

Table 1 Proposed CNN-FPS for ASC.

Modules	Description
Input	/
FPS module	Ln=[[L^0, L^1,..., L^n]]
Conv_1	Kernal:$3 \times 3, 32$
Conv_2	Kernal:$3 \times 3, 32$
Conv_3	Kernal:$3 \times 3, 32$
AVG Pooling_1	Kernal:2×4, $stride : [2, 4]$
Conv_4	Kernal:$3 \times 3, 32$
Conv_5	Kernal:$3 \times 3, 32$
AVG Pooling_2	Kernal:2×4, $stride : [2, 4]$
Conv_6	Kernal:$3 \times 3, 64$
Conv_7	Kernal:$3 \times 3, 64$
Conv_8	Kernal:$3 \times 3, 64$
Conv_9	Kernal:$3 \times 3, 64$
Conv_10	Kernal:$1 \times 1, 10$
Fusion	/
SoftMax	10

3.2 Experiment

To test the performance of our proposed feature pyramid segmentation module, we conducted multiple comparative experiments to test different feature segmentation strategies.

- baseline CNN: excluding any feature segmentation strategy, that is, the segmentation strategy of L^0 in the FPS module.
- CNN-FPS(L^x): a single pyramid pooling strategy, where $x = 1$, 2, or 3. For instance, CNN-FPS(L^1) represents dividing the input feature into 2×2 local features of the same scale as the input for a single CNN.
- CNN-FPS-L1: a CNN network containing L^0 and L^1 segmentation strategies, that is, a total of two sets of CNN modules, with inputs of the original spectrum and four sets of segment-level features segmented by L^1 strategy as inputs for another CNN path.
- CNN-FPS-L2: a CNN network containing L^0, L^1, and L^2 segmentation strategies, consisting of three sets of CNN modules. The inputs for each CNN path are the original spectral map features, four sets of segment-level features produced by L^1 strategy, and sixteen sets of segment-level features generated by L^2 strategy.
- CNN-FPS-L3: a CNN network containing L^0, L^1, L^2, and L^4 segmentation strategies, with four CNN paths. The input of each path is the segment-level features generated by each FPS strategy. In addition, L4 strategy represents dividing input features into continuous local features of 8×8 segments.

The test results of all models are shown in Table 2.

First off, the single-scale feature maps generated by the FPS module might not be sufficient to improve the performance of the baseline CNN. CNN-FPS(L^1) improves the baseline system's performance on DCASE 2018, but other strategies may reduce the performance. Additionally, the model's performance may decline as the input feature map's size diminishes; for instance, CNN-FPS(L^4) performs far

Table 2 Performance (%) comparison of different modules.

module	DCASE2018	DCASE2019	DCASE2021
Baseline CNN	75.24	76.64	63.98
CNN-FPS(L^1)	76.33	76.63	62.06
CNN-FPS(L^2)	74.74	76.37	63.31
CNN-FPS(L^4)	71.32	71.76	61.32
CNN-FPS-L1	77.04	77.47	65.73
CNN-FPS-L2	77.56	78.73	65.90
CNN-FPS-L3	76.05	77.53	65.20

Table 3 Overall results on DCASE 2018 and 2019.

System	2018(%)	2019(%)	2021(%)
SubSpectralNet [6]	74.08	/	/
MCTA-CNN [7]	72.40	75.71	/
Atrous-CNN [8]	72.7	/	/
ResNet [9]	/	75.1	/
DCASE Baseline [5]	59.7	62.5	46.90
Zeinali_BUT [10]	70.3	/	/
Liang_HUST [11]	/	70.7	/
Kek_NU [12]	/	70.7	63.0
Galindo-Meza_ITESO [13]	/	70.7	53.9
CNN-FPS	77.56	78.73	65.90

worse than CNN-FPS(L^1). It is essential to consider multi-scale features since input feature maps of various sizes contain time-frequency information of various scales.

According to the statistics presented in Table 2, it is evident that CNN-FPS-L2 exhibits superior performance. The recognition rates for CNN-FPS-L2 demonstrate improvements of 2.32%, 2.09%, and 1.75% compared to the baseline CNN. Furthermore, it is evident from the table that the model's performance exhibits a pattern of initial improvement followed by a decline when the number of feature segmentation increases. This phenomenon may arise when the initial segmentation of features is insufficient in quantity, resulting in a lack of refinement in the identified features. When the quantity of original feature segmentation is great, it might lead to the degradation of short-term time-frequency information within the features.

In addition, we also compared and analyzed CNN-FPS with some CNN-based models. The comparative results of all experiments are shown in Table 3. The comparison model consists of CNN-based ASC tasks and DCASE tasks, including “SubSpectralNet”, “MCTA-CNN”, “Atrous-CNN”, “ResNet”, “Zeinali_BUT”, “Liang_HUST”, “Kek_NU”, “Galindo-Meza_ITESO” and “DCASE Baseline”. The results demonstrate that the proposed CNN-FPS model has superiority over alternative CNN-based models. Compared to the DCASE baseline, the suggested model exhibits an increase in recognition rate of 17.86%, 16.23%, and 19% correspondingly.

4. Conclusion

This article proposes a multi-scale feature pyramid segmentation strategy based on CNN. The CNN-FPS model utilizes the log-mel spectrum as its input, while the feature pyramid segmentation layer is meant to conduct multi-scale segmentation on the input log-mel spectrum. This layer uses various CNN paths to capture time-frequency information at different scales. Subsequently, the CNN backend integrates all features as the model's output. The model's efficacy was assessed during the experimental phase by testing the proposed feature pyramid segmentation approach. Compared to other works based on ASC, the developed model exhibits superior recognition performance.

References

[1] N.W. Hasan, A.S. Saudi, M.I. Khalil, and H.M. Abbas, “A genetic algorithm approach to automate architecture design for acoustic scene classification,” IEEE Trans. Evol. Comput., vol.27, no.2, pp.222–236, April 2023.

[2] A. Madhu and S. K, “‘RQNet: Residual quaternion CNN for performance enhancement in low complexity and device robust acoustic scene classification,” IEEE Trans. Multimed., vol.25, pp.8780–8792, 2023.

[3] H. Song, S. Deng and J. Han, “Exploring inter-node relations in CNNs for environmental sound classification,” IEEE Signal Process. Lett., vol.29, pp.154–158, 2022.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intel., vol.37, no.9, pp.1904–1916, 2015.

[5] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), 9–13 Nov. 2018.

[6] S.S. R Phaye, E. Benetos, and Y. Wang, “SubSpectralNet – Using sub-spectrogram based convolutional neural networks for acoustic scene classification,” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp.825–829, 2019.

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

[8] Z. Ren, Q. Kong, J. Han, M.D. Plumbley, and B.W. Schuller, “Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes,” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.56–60, 2019.

[9] A.M. Tripathi and A. Mishra, “Self-supervised learning for environmental sound classification,” Applied Acoustics, vol.182, 108183, 2021.

[10] H. Zeinali, L. Burget, and H. Cernocky, “Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge,” DCASE2018 Challenge, Tech. Rep., Sept. 2018.

[11] H. Liang and Y. Ma, “Acoustic scene classification using attention-based convolutional neural network,” DCASE2019 Challenge, Tech. Rep., June 2019.

[12] X.Y. Kek, C.S. Chin, and L. Ye, “Technical paper: Deep scattering spectrum with mobile network for low complexity acoustic scene classification,” DCASE2021 Challenge, Tech. Rep., June 2021.

[13] C.A. Galindo-Meza, J.A. del Hoyo Ontiveros, J.I. Torres Ortega, and P. Lopez-Meyer, “End-to-end CNN optimization for low-complexity acoustic scene classification in the DCASE 2021 challenge,” DCASE2021 Challenge, Tech. Rep., June 2021.