ACOUSTIC SCENE CLASSIFICATION BASED ON DEEP CONVOLUTIONAL NEURAL NETWORK WITH SPATIAL-TEMPORAL ATTENTION POOLING

Technical Report

Huang Zhenyi, Jiang Dacan

School of Computer, South China Normal University
Guangzhou, China
767163703@qq.com, kevien2007@foxmial.com

ABSTRACT

Acoustic scene classification is a challenging task in machine learning with limited data sets. In this report, several different spectrograms are applied to classify the acoustic scenes using deep convolutional neural network with spatial-temporal attention pooling. In addition, mixup augmentation is performed to further improve the classification performance. Finally, majority voting is performed on six different models and an accuracy of 73.86% is achieved which is 11.36 percentage points higher than the one of the baseline system.

***Index Terms*—** Acoustic scene, CNN, MFCC, CQT, Spatial-temporal attentione

1. INTRODUCTION

Sound contains complex information and plays an important role in human life. Acoustic scene classification has become one of the main tasks of sound analysis. Nowadays, acoustic scene classification has been applied in monitoring, robot navigation and context-aware services.

Spatial-temporal attention, which was used in video-based research in the early days[1]. It can be divided into two categories, the spatial attention and temporal attention, which acquire better features in time domain and space domain respectively. In[1], spatial-temporal attention was used to improve the accuracy of video description and in[2], spatial-temporal attention was used to improve the accuracy of Person Re-identification based on videos. In[3] spatial-temporal attention is applied in acoustic scene classification as well. Attention layers learn attention weight vectors in the spatial and temporal dimensions from the recurrent output, collectively constructing a spatio-temporal attention mask to weigh and pool the recurrent output into a single feature vector for classification. According to their experiment results, the proposed method achieved good classification performance in the LITIS Rouen dataset. Inspired by this work, we introduce spatial-temporal attention pooling into our proposed solution.

2. DATA PREPARATION

The method of generating the spectrograms refers to the technical report in DCASE-2017 Challenge[4]. We use three channels for the audios: the left channel, the right channel and the mixed channel (averaging the left channel and the right channel).

This work was partially supported by the Educational Commission of Guangdong Province, China under Grant 2016KTSCX025.

2.1. MFCC Spectrogram

MFCC spectrograms are used in our method. The feature extraction of Mel frequency cepstrum coefficients includes two key steps: transforming to Mel frequency, and performing cepstrum analysis. To extract MFCC features, we use python the Librosa library. The parameter settings are as followed: the sample rate is set as 44100 and the number of Mel coefficients is 128.

Firstly, short-time Fourier transformation(STFT) is performed to transform the time domain signal into time-frequency presentations. The window function of STFT is a hann window. During the generation of MFCC spectrograms, we use two different window sizes, 2048 and 1024, and the hop sizes are 1024 and 512, respectively.

Once the spectrogram has been generated, we split it into several smaller patches with fixed width and shift length. We use two kinds of MFCC patches in our experiments. For the first one, the patch width is 128 pixels and the shift step is 32, and the second one, the patch width is 128 and the shift step is 64. We resize every patch into 128×128. Finally, for each MFCC spectrogram, 11 patches can be generated. As a result, we can generate 33 segments from a single audio file, corresponding to three channels.

2.2. CQT Spectrogram

Similarly, Constant Q-transform (CQT) is used to convert time domain signal into time-frequency representation, which was originally used for music recognition. Music is different from ordinary sounds. It is dense in low frequency and sparse in high frequency. To deal with this problem, CQT processes audio with variable resolutions. That is, selecting more sampling points in low frequency and fewer sampling points in high frequency.

The CQT spectrogram is generated on the CQT features which are computed from the raw audio frames by using the python Librosa library. When invoking the CQT function in the library, the sampling rate is set as 44100 and filter scale is set as 4; hop length is 512 and the number of bins is 110. For each audio file, we generate three CQT spectrograms (the width is 938, the height is 110), each for a channel. Once again, we split the spectrogram into patches. The patch width is 128 and the shift step is 64.

2.3. HPSS-MFCC Spectrogram

Inspired by [5], we also use Harmonic-percussive source separation to deal with the audio before generating MFCC Spectrogram. On a very rough level, many sounds can be divided into two categories:

Table 1: DCNN model with Spatial-temporal attention pooling

Input $1 \times 128 \times 128$
3×3 Conv(pad-'SAME', stride-2)-64-BN-ReLu
3×3 Conv(pad-'SAME', stride-1)-64-BN-ReLu
2×2 MaxPooling(stride-2,1)
3×3 Conv(pad-'SAME', stride-1)-128-BN-ReLu
3×3 Conv(pad-'SAME', stride-1)-128-BN-ReLu
2×2 MaxPooling(stride-2,1)
3×3 Conv(pad-'SAME', stride-1)-256-BN-ReLu
3×3 Conv(pad-'SAME', stride-1)-256-BN-ReLu
1×1 Conv(pad-'SAME', stride-1)-256-BN-ReLu
2×2 MaxPooling(stride-2,1)
3×3 Conv(pad-'SAME', stride-1)-512-BN-ReLu
3×3 Conv(pad-'SAME', stride-1)-512-BN-ReLu
1×1 Conv(pad-'SAME', stride-1)-512-BN-ReLu
2×2 MaxPooling(stride-2,1)
Spatial-Temporal Attention pooling
Fully connected layer-10-ReLu
10-way SoftMax

harmonic or percussive sounds. The goal of harmonic-percussive source separation (HPSS) is to decompose a given input signal into a sum of two component signals, one consisting of all harmonic sounds and the other consisting of all percussive sounds. In our experiment, the method applied to separate harmonic source and percussive source is the same to the one of [5].

3. NETWORK ARCHITECTURE

3.1. Deep Convolutional Neural Network

Similar to [4], we adopt a DCNN model to perform acoustic scene classification, which follows a VGG style network. Our network architecture is depicted in Table 1. After multiple layers of convolution and maxpooling, we add the spatial-temporal attention pooling layer.

3.2. Spatial-Temporal Attention pooling

As shown in Table 1, after four layers of convolution and pooling, we obtain the output $\theta \in R^{4\times64\times512}$. Then θ is reshaped to $O \in R^{S\times T}$, where $S = 4 \times 512$ and $T = 64$. S is considered as spatial domain and T is considered as temporal domain.

After O is obtained, we have to learn a spatial-temporal attention mask to pool and reduce it into a single feature vector. $a^{tem} \in R^T$ and $a^{spa} \in R^S$ are two attention vectors, where a^{tem} for temporal attention and a^{spa} for spatial attention.[3] We have the following formula:

$$a_t^{tem} = \left(\frac{exp(f(o_t))}{\sum_{i=1}^{T} exp(f(o_i))} \right), \tag{1}$$

$$a_s^{spa} = \left(\frac{exp(\bar{f}(\bar{o}_t))}{\sum_{i=1}^{S} exp(\bar{f}(\bar{o}_i))} \right), \tag{2}$$

In (1) and (2), a_t^{tem} is the temporal attention weight at the time index t, $1 \le t \le T$.Similarly, a_s^{spa} is the spatial attention weight at the spatial index s, $1 \le s \le S$. o_t represents the column of O at the column (i.e. temporal) index t and $\bar{o}_s$ represents the row of O at the row (i.e. spatial) index s. f and $\bar{f}$ are the scoring functions of temporal and spatial attention layers, and they can be computed as:

$$f(o) = o^T W, \tag{3}$$

$$\bar{f}(\bar{o}) = \bar{o}^T \bar{W}, \tag{4}$$

In(3) and (4), W and $\bar{W}$ are the trainable weight matrices.

$$A = a^{spa} \otimes a^{tem}, \tag{5}$$

In (5), we obtain the spatial-temporal attention mask A by vector outer product between a^{spa} and a^{tem}

$$x_s = \sum_{t=1}^{T} tanh(A_{st} O_{st}), s \in [1, S] \tag{6}$$

Finally, We get the feature vector $x \in R^S$ in (6). we do element-wise multiplication between the output O and the spatial-temporal attention mask A. Next, use a tanh activation prior to the summation.Inspired by[6], because of the output x range(-1,1), it is said that tanh activation does not only suppress the irrelevant features but also enhances the informative ones in the resulting feature vector x. In the end, the obtained feature vector x will be sent to a fully Connected layer to complete classification.

4. DATA AUGMENTATION

Because of the insufficient training data, it is necessary to expand the training data, which is expected to improve the generalization ability of the model. Mixup[7] is an effective data augmentation method proposed in 2017. Mixup had proposed a general augmentation approach: mixing different samples of the training set according to their weights, and mixing labels according to their weights.The method is as follows:

$$X = \lambda X_i + (1 - \lambda) X_j \tag{7}$$

$$y = \lambda y_i + (1 - \lambda) y_j \tag{8}$$

Where, $\lambda \in [0, 1]$ and it is acquired by sampling from the beta distribution $Beta(\alpha, \alpha), \alpha \in (0, \infty)$. Note that α is a hyper parameter. Xi and Xj are different data, yi and yj are their corresponding label. In our experiment, we use the mixup to augment the MFCC and HPSS-MFCC spectrograms.

5. PROPOSED METHOD

In our experiment, we have totally trained six models. From the Figure 1, we totally used three different methods to generate spectrogram. When generating MFCC spectrogram, two different window sizes and hop sizes are used. The first one is 2048 window size and 1024 hop size, the second one is 1024 window size and 512 hop size. When generating the HPSS-MFCC spectrograms, the first setting is applied. Consequently, we totally have four different types of spectrograms. They are $\text{MFCC}_{2048,1024}$, $\text{MFCC}_{2048,1024}$, CQT, $\text{HPSS-MFCC}_{2048,1024}$. We trained six models with two similar networks, one having a spatial-temporal attention pooling layer,

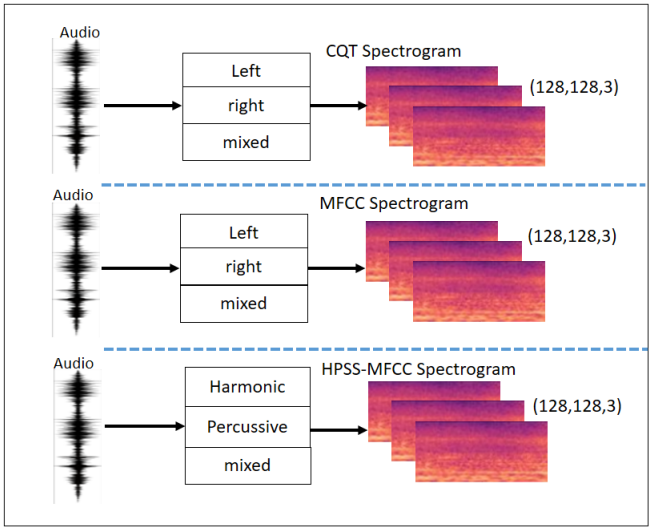

Figure 1: Spectrogram Generation Method

while the other one having not. Except for the spatial-temporal attention pooling layer, the two networks are totally identical. The one with the spatial-temporal attention pooling layer is denoted as DCNN^{sp} in the remainder of this report, and the other one is denoted as DCNN^{nonsp}. The six models are as follows:
(1) MFCC model: MFCC$_{2048,1024}$ is fed into DCNN^{nonsp} as input.
(2) MFCC-STAP model: MFCC$_{2048,1024}$ is fed into DCNN^{sp} as input.
(3) MFCC-STAP-mixup model: Mixup augmentation is performed on the MFCC$_{2048,1024}$ data and the DCNN^{sp} is applied upon them.
(4) CQT+MFCC model: CQT and MFCC$_{1024,512}$ are organized as two channels which are fed into the DCNN^{nonsp} network.
(5) CQT+MFCC-STAP model: CQT and MFCC$_{1024,512}$ are organized as two channels which are fed into the DCNN^{sp} network.
(6) MFCC(HPSS)-STAP-mixup model: Mixup augmentation is performed on the HPSS-MFCC$_{2048,1024}$ data and the DCNN^{sp} is applied upon them.

As shown in Figure2, we integrate 6 models using majority vote method upon each sample probability.

6. EXPERIMENTS AND RESULT

6.1. Datasets

The dataset for this task is the TAU Urban Acoustic Scenes 2019 dataset, consisting of recordings from various acoustic scenes. This dataset extends the TUT Urban Acoustic Scenes 2018 dataset with other 6 cities to a total of 12 large European cities. For each scene class, recordings were done in different locations; for each recording location there are 5-6 minutes of audio. The original recordings were split into segments with a length of 10 seconds that are provided in individual files. Available information about the recordings include the following: acoustic scene class, city, and recording location. The dataset includes 10 scenes which are Airport, Indoor shopping mall, Metro station, Pedestrian street, Public square, Street with medium level of traffic, Travelling by a tram, Travelling by a bus, Travelling by an underground metro, Urban park. In order to improve the accuracy of the experiment, we use 5-fold cross validation in our experiment.

6.2. Experimental parameters

In the experiment, the initial learning rate was set as 0.0001, and the batch size was set as 256. In the training process, we set the number of epochs as 100. In order to accelerate training, we used early stop strategy. Specifically, if the accuracies of 20 consecutive epochs did not improve, the training will be terminated in advance. We also use the L2-Regularization with a weight decay of 0.0005.

Note that for the CQT+MFCC and CQT+MFCC-STAP models, there are several differences of network parameters from the ones presented in Table 1. Firstly, the input of the network is $2 \times 128 \times 128$, where CQT and MFCC act as one channel, respectively. Second, the first convolution stride in the first layer is set as 1, and the output channels of all convolutions in the third layer are set to 256. As a result, we gain an $O \in R^{4\times128\times256}$ as the input of the spatial-temporal attention pooling layer.

6.3. Ensemble method

Majority voting is a straightforward and simple method in this situation. Specifically, each sample produces one vote and the class which wins the most votes is considered as the final result. As shown in Table 2, with the ensemble of six models, an accuracy of 73.86% is achieved for the TAU Urban Acoustic Scenes 2019 dataset.

6.4. Result

Experimental results are demonstrated in Table 2. As we can see, all the results have outperformed, the one of the baseline system. In addition, by ensemble the six models, the accuracy is greatly improved (which is 73.8% here). Table 3 describes the per scene accuracy in audio in the best model.

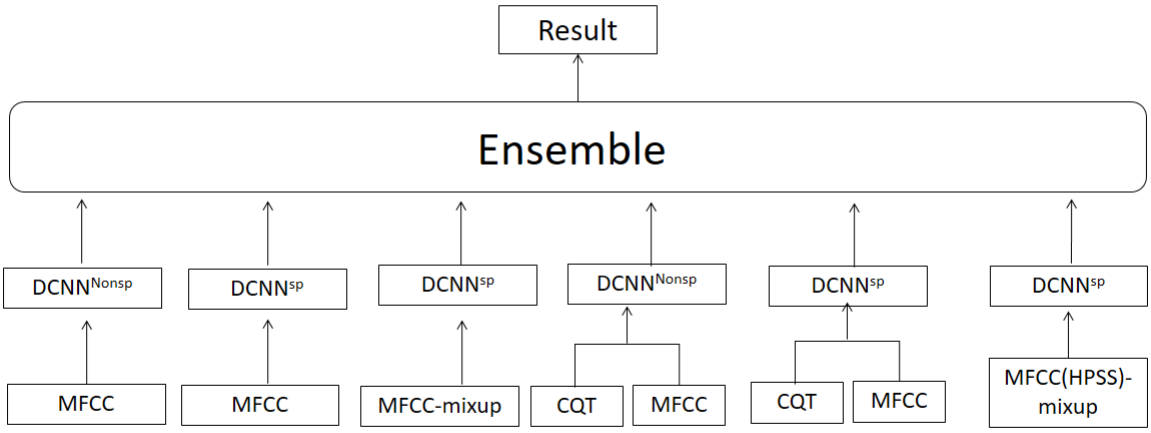

Figure 2: The ensemble method

Table 2: Classification results of Development dataset. STAP: Spatial-Temporal Attention pooling. HPSS: Harmonic-percussive source separation

Models	Accuracy(Development)
Baseline	62.5% ($\pm$0.6)
MFCC	63.2% ($\pm$0.5)
MFCC-STAP	69.6 % ($\pm$0.5)
MFCC-STAP-mixup	71.1% ($\pm$0.5)
CQT+MFCC	68.8% ($\pm$0.5)
CQT+MFCC-STAP	68.1% ($\pm$0.5)
MFCC(HPSS)-STAP-mixup	65.5% ($\pm$0.5)
Ensemble model	73.86%

Table 3: classification accuracy of per scene

Scene label	Baseline	Ensemble accuracy
Airport	48.4%	78.3%
Bus	62.3 %	91.0%
Metro	65.1%	67.1%
Metro station	54.5%	67.1%
Park	83.1%	90.7%
Public square	40.7%	54.2%
Shopping mall	59.4%	69.2%
Street, pedestrian	60.9 %	61.0%
Street, traffic	86.7 %	91.1%
Tram	64.0 %	67.9%

7. CONCLUSION

In this paper, we use a variety of spectrograms(such as MFCC, CQT, and HPSS-MFCC spectrograms) to classify scenes. In addition, we use the spatial-temporal attention pooling to promote the classification performance. According to our experimental results,the accuracy of single best model is 71.1%,which is 8.6 percentage points higher than that of the baseline. However, when ensemble model is considered, the best result is 73.86%, which is 11.36 percentage points higher than that of baseline.

8. REFERENCES

[1] Y. Tu, X. Zhang, B. Liu, and C. Yan, "Video description with spatial-temporal attention," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1014–1022.

[2] S. Rao, T. Rahman, M. Rochan, and Y. Wang, "Video-based person re-identification using spatial-temporal attention networks," *arXiv preprint arXiv:1810.11261*, 2018.

[3] H. Phan, O. Y. Chén, L. Pham, P. Koch, M. De Vos, I. McLoughlin, and A. Mertins, "Spatio-temporal attention pooling for audio scene classification," *arXiv preprint arXiv:1904.03543*, 2019.

[4] Z. Weiping, Y. Jiantao, X. Xiaotao, L. Xiangtao, and P. Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," in *Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017.

[5] M. Müller, "Harmonic percussive source separation."

[6] C. Zhu, X. Tan, F. Zhou, X. Liu, K. Yue, E. Ding, and Y. Ma, "Fine-grained video categorization with redundancy reduction attention," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 136–152.

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