LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION USING PRIMARY AMBIENT EXTRACTION AND CYCLEGAN

Technical Report

Haocong Yang[1], *Chuang Shi*[2], *Huiyong Li*[3]

University of Electronic Science and Technology of China, Chengdu, China
[1] yanghaocong@std.uestc.edu.cn
[2] shichuang@uestc.edu.cn
[3] hyli@uestc.edu.cn

ABSTRACT

This report describes our submissions for DCASE2020 Challenge task 1b (Low-Complexity Acoustic Scene Classification). In each submission, constant-Q transform is used as acoustic feature, and the corresponding classifier is a full convolution neural network based on residual blocks. The classifier parameters use half-precision (16 bit) float-point number to limit the model size and accelerate training. We use primary ambient extraction in the audio front-end processing, and generate virtual samples according to the phase information of binaural audio. These virtual samples will be used for one of the submissions. We also used the virtual samples generated by CycleGAN for another submission. Finally, we give a 4-fold cross validation submission that meets the complexity limit. The highest macro recognition accuracy of the above methods in the development dataset is 96.05%, and the log loss is 0.120.

Index Terms— Acoustic scene classification, convolutional neural network, primary ambient extraction, CycleGAN.

1. INTRODUCTION

DCASE has been held six times since 2013 [1]. Acoustic Scene Classification (ASC) has been one of the resident projects since the first session. It is also the main task for the aspect of acoustic scene [2]. In previous challenges, many creative research methods of ASC have emerged. Under the competition iteration, ASC has gradually concentrated on specified perspectives derived from different real-world requirements and divided into some subtasks, including ASC system transfer across cities and countries (2018task1a, 2019task1a), ASC judgment in the face of unknown classification (2019task1c), ASC system transfer on different devices (2018task1b, 2019task1b, 2020taska). As the task 1b in 2020, ASC system in low complexity is the topic of competition [3]. The following descriptions of the report is all refer to this task.

There are two datasets involved in task1b, including TAU Urban Acoustic Scenes 2020 3Class Development dataset and TAU Urban Acoustic Scenes 2020 3Class Evaluation dataset. The recording equipment of the datasets is soundman OKM II Klassik / Studio A3 and zoom F8. During the recording process, the microphone will be worn on the right and left ears of the collector to restore the working mode of the human auditory system, which also makes the binaural audio samples of datasets contain phase information. The TAU Urban Acoustic Scenes 2020 3Class Development dataset is a high-quality binaural audio dataset, which contains a variety of sound scene samples collected in 10 European cities [4]. The total recording time is 40 hours, and data are respectively recorded in 10 surrounding, including airport, travelling by a bus, travelling by an underground metro, metro station, urban park, public square, indoor shopping mall, pedestrian street, street with medium level of traffic, and tram, each scene corresponding to 1440 recording samples. Total 14400 samples with 10 seconds were labeled in three classifications with indoor (airport, shopping mall, and metro station), outdoor (street pedestrian, public square, street traffic, and urban park), transportation (bus, tram, and metro). The TAU Urban Acoustic Scenes 2020 3Class Evaluation dataset does not disclose label information and is only used for the final evaluation of task1b.

This technical report describes the methods we used for the 4 submissions in task1b. The first submission uses CQT and CNNs, combined with some machine learning methods. Submission 2 combines PAE on the basis of submission 1, submission 3 uses CycleGAN based on submission 1, and submission 4 uses 4-fold cross validation on the basis of submission 1. Chapters 2 to 3 describes our research methods, and chapter 4 describes our experimental settings. Results and final submission are demonstrated in chapter 5.

2. DATA PREPROCESSING

2.1. Acoustic Feature

We selected CQT as acoustic feature. For a single binaural audio sample with length of 10 s, we firstly resample it to 22.05khz, and calculate the absolute value of audio raw waveform’s mean, which was taken as the reference value to scale the waveform. On this basis, 1024 sampling points are used as steps to divide the audio into 216 frames. In constant Q transformation, C0 is set as the lowest tone, number of bins is 336, with 36 bins per octave. Finally, the energy representation of CQT is transformed to dB scale, and the acoustic feature with the shape of (2, 336, 216) are obtained.

Table 1: The structure of proposed classifier. Where ch denote number of channels, ksize denote size of filter, and psize denote pooling size. N=36 in submission 1-3, and N=22 in submission 4.

Stage	Settings
Input	Input 2×336×216
	Conv2d (ksize=5, pading=2, stride=2, ch= N)
	BatchNorm2d (ch= N)
	ReLU (ch= N)
Transformation	ResidualBlock (ksize1=3, ksize2=1, ch= N)
	MaxPooling (psize=2)
	ResidualBlock (ksize1=3, ksize2=1, ch= N)
	MaxPooling (psize=2)
	ResidualBlock (ksize1=3, ksize2=1, ch= N)
	MaxPooling (psize=2)
	ResidualBlock (ksize1=1, ksize2=1, ch= N×2)
	ResidualBlock (ksize1=1, ksize2=1, ch= N×2)
	ResidualBlock (ksize1=1, ksize2=1, ch= N×2)
Output	Conv2d (ksize=1, padding=2, stride=2, ch=3)
	BatchNorm2d (ch=3)
	GlobalAveragePooling2d (ch=3)
	Output 3-way SoftMax

2.2. Primary Ambient Extraction

The original purpose of the primary ambient extraction is to solve the problem of device mismatch when the channel format audio signal is played back [5]. Based on the simplified signal model, the primary ambient extraction algorithm constructs a mathematical underdetermined problem, and then uses the sparsity constraints of the primary components to determine the unique solution. When processing stereo recording, the algorithm will divide it into four channels, which are the left primary component and ambient component, and the right primary component and ambient component. The decomposition results ensure the maximum correlation between the left and right primary components, the maximum sparsity of the primary components in the whole time-frequency domain, and the energy balance of the left and right ambient components. We restructure the audio samples and generate virtual audios with PAE, the detail of PAE can be found in He's paper [6].

In 2019 task1a submission, we used PAE to extract new four channel features and integrate them into our classification system [7]. Although the complexity of the feature is higher, and we have not adjusted the training parameters for the feature, the final result is still improved. This makes us believe that there are still some exploitable parts of stereo audio phase information in ASC. Therefore, we continue to use PAE in this year's competition, and try to improve the defects in the previous methods, finding a more efficient way of utilization.

Table 2: The structure of generator in CycleGAN.

Stage	Settings
Input	Input 2×336×216
	Conv2d (ksize=7, padding=2, stride=1, ch=64)
	BatchNorm2d (ch=64)
	ReLU (ch=64)
Encoding	Conv2d (ksize=3, padding=1, stride=2, ch=128)
	BatchNorm2d (ch=128)
	ReLU (ch=128)
	Conv2d (ksize=3, padding=1, stride=2, ch=256)
	BatchNorm2d (ch=256)
	ReLU (ch=256)
Transformation	ResidualBlock (ksize1=3, ksize2=3, ch=256)
	ResidualBlock (ksize1=3, ksize2=3, ch=256)
	ResidualBlock (ksize1=3, ksize2=3, ch=256)
Decoding	ConvTranspose2d (ksize=3, padding=1, stride=2, ch=128)
	BatchNorm2d (ch=128)
	ReLU (ch=64)
	ConvTranspose2d (ksize=3, padding=1, stride=2, ch=64)
	BatchNorm2d (ch=64)
	ReLU (ch=64)
Output	Conv2d (k_size=7, padding=3, stride=1, ch=2)
	Tanh(ch=2)
	output 2×336×216

2.3. CycleGAN

In task 1b, due to the limitation of model complexity, optimization for a specific classifier should be considered. For this reason, we use GAN to generate more virtual samples form confusing classification data of current classifier. GAN includes generator and discriminator. Generator is responsible for imitating the real samples and hide generated samples into the real dataset, while discriminator is trying to separate the fake samples from the real dataset. After the game between this two, the forgery technology of generator is more and more powerful, and the identification technology of discriminator is also improved. Until discriminator can no longer tell whether the data is real or fake, the confrontation process reaches a dynamic balance [8]. CycleGAN consists of two symmetrical GANs, forming a ring network. Two GANs share two generators and each has a discriminator, so there are two discriminators and two generators [9]. CycleGAN can generate more useful samples without labeled data.

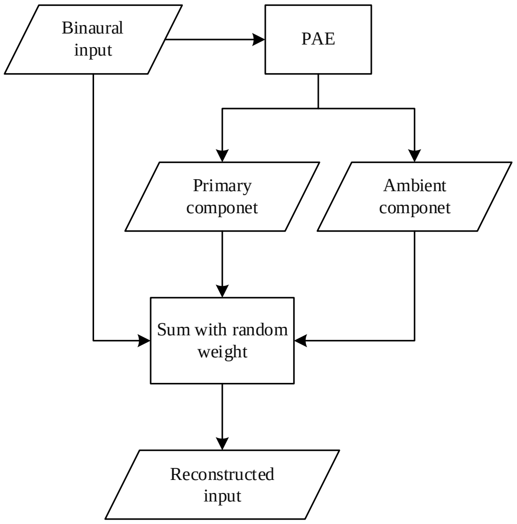

Figure 1: Reconstruction samples by PAE

2.4. Mixup

Mixup method is a form of neighborhood risk minimization [10]. It makes a higher sampling rate near the target optimization point of feature hyperplane, and the final optimized location will more likely close to the best. It sums two randomly selected features with a weight to generate new a virtual feature, and the corresponding labels of the two features are similarly operated. This process is expressed as:

$$\begin{aligned}\tilde{x} &= x_i + \left(1-\lambda\right)x_j \\ \tilde{y} &= y_i + \left(1-\lambda\right)y_j,\end{aligned} \qquad (1)$$

x_i and x_j are two randomly selected features; y_i and y_j are corresponding loss functions. The random variable λ follows the beta distribution $Be(\alpha,\alpha)$.

3. STRUCTURE

Convolutional neural network reduces the complexity of the network model through three strategies: local receptive field, weight sharing and down sampling [11]. It has excellent advantages in translation, rotation and scaling since its distortion variance. Therefore, it is widely used in image classification, target recognition, speech recognition and other fields.

The structure of our classifier is a full convolution neural network. As shown in the table 1. Its main part consists of residual block [12], including an input stage, six residual block and an output stage. The output of the first three residual blocks use 2×2 maxpooling to reduce the complexity of features. In the output stage, result of global average pooling and softmax is directly used before output.

The structure of generator in CycleGAN is shown in table 2, the number of residual blocks in the transformation stage is set to 3. For discriminator, we use the same structure provided in the original paper [9].

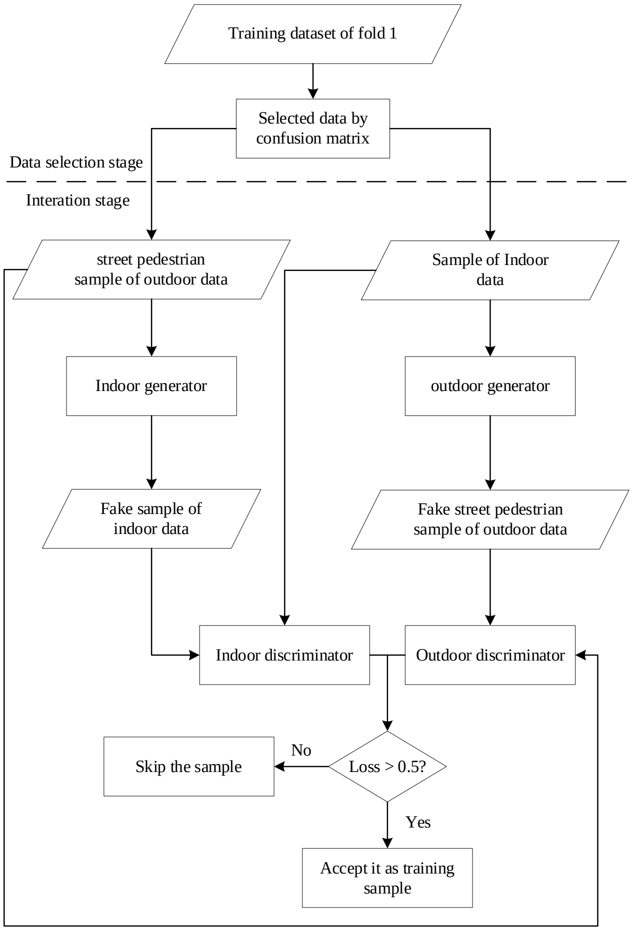
Training dataset of fold 1
Selected data by confusion matrix
Data selection stage
Interation stage
street pedestrian sample of outdoor data
Sample of Indoor data
Indoor generator
outdoor generator
Fake sample of indoor data
Fake street pedestrian sample of outdoor data
Indoor discriminator
Outdoor discriminator
Skip the sample
No
Loss > 0.5?
Yes
Accept it as training sample

Figure 2: The use strategy of the CycleGAN, taking the street pedestrian as an example.

4. EXPERIMENT SETTING

In the local experiment of submitting 1-3, we randomly divided 15% data from the training dataset of fold1 as the validation dataset. In submission 4, we further limited the scale of a single model to conduct 4-fold cross validation, ensuring a submission with no over fitting on data distribution. In the final submission stage, our training data changed to the whole development dataset, and other configurations continue to follow the local experiment.

As for the parameter usage, all submissions have used Mixup, where the hyper-parameter alpha is set to 0.2. In submission 2, we used PAE to reconstruct the audio samples, and the specific process is shown in Figure 1. In the third submission, we selected a part of data that would cause high confusion by analysing the confusion matrix of submission 1. Based on two sets of data, we trained CycleGAN to obtain reasonable samples, and qualified samples will join our training data. The specific strategy of the CycleGAN is shown in Figure 2. The data format used in the whole training process is half-precision (16 bit) float-point number. Because of the scaling of the waveform in the preprocessing stage, the dB expression of signal is converted, so that we can avoid numerical problems and accelerate the convergence. Adaptive moment estimation (Adam) algorithm was used as the optimizer, whose learning rate, betas, eps are set to 0.001, (0.9, 0.999), and 0.001, respectively. In the last 15 epochs, the learning rate was set to minimum half-precision float-point positive value, so that the result is closer to the optimal goal.

5. RESULTS AND SUBMISSIONS

5.1. Results on development dataset

The results of the four methods are shown in Table 3, in which the results of system 1-3 use the average score of four times training. System 1 is the result of applying CQT directly to CNNs. System 2 is the result of using PAE. System 3 is the result of using CycleGAN, and system 4 is the result of 4-fold cross validation.

In system 2, after the real recording audio is separated based on PAE signal model, the energy distribution of the component is obviously different from the original sample. We think it will count against the classifier to learn the statistical law. Therefore, we combine a small weight of PAE generated component with the original sample, and use the correlation between component and the original sample to disturb the original sample, further improving the variety of data. In times of local experiment, system 2's score is superior to that of system 1 in both mean and variance.

In system 3. Thanks to the recording location and environment information attached to the dataset in addition to labels, we can further narrow the search scope for confusing data. It provides great convenience for CycleGAN to establish symmetric datasets.

Because our experiments are all carried out under the Pytorch framework, the complexity of the model is calculated without using the script provided by challenge committee. In system 1-3, we limited the parameters within 120000 (calculated through the API of Pytorch), and used half-precision float-point numbers to ensure that the complexity of the models within the restriction.

Submission 4 is an attempt to detect the over fitting level on data distribution. The total number of parameters exceeds 125000, but the size of the four sub models still does not exceed 500K (pytorch models stored in .pkl format) after using half-precision float-point numbers.

Table 3: Result of local experiments.

Sys ID	Total parameters	Model size	Macro-average accuracy (%)	Log loss
1	119382	258 kB	94.92	0.237
2	119382	258 kB	95.86	0.200
3	119382	258 kB	96.05	0.187
4	45,546×4	112kB×4	91.95	0.305

5.2. Submission

For final submission, the training data changed to the whole development dataset, and other configurations still to follow the local experiment., they are submitted as:

1. **Yang_UESTC_task1b_1:** This submission is the result of applying CQT directly to CNNs, with 15% of data used as validation dataset (abbreviation: CNNs).
2. **Yang_UESTC_task1b_2:** This submission is the result of applying PAE based on submission 1, with 15% of data used as validation dataset (abbreviation: CNNs_PAE).
3. **Yang_UESTC_task1b_3:** This submission is the result of applying CycleGAN based on submission 1, with 15% of data used as validation dataset (abbreviation: CNNs_Cyc).
4. **Yang_UESTC_task1b_4:** This submission is the result of applying 4-fold cross validation based on submission 1, with 15% of data used as validation dataset (abbreviation: CNNs_4CV).

6. CONCLUSION

In this paper, because of complexity restriction of task1b, we pay attention on data preprocessing. We used PAE and CycleGAN to conduct data augmentation, and also detected the over fitting level on data distribution. The result of local experiment demonstrate that those methods can further improve the performance of classifier. In addition, because the half-precision float-point parameters are used in the model of submission 1-3, the size of the model can be further increased to enhance the performance. However, considering the compatibility problems of the half-precision float-point parameters in some situations, we still choose the numbers of parameters as the primary complexity restriction. The best method's Macro-average accuracy in development dataset is 96.05%, and the log loss is 0.120.

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] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification: an overview of DCASE 2017 challenge entries," in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, Tokyo, Japan, 2018.

[3] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in proceedings of the *Detection and Classification of Acoustic Scenes and Events 2020 Workshop* (*DCASE2020*). 2020. Submitted.

[4] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *the 2018 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2018.

[5] L. Chen, C. Shi, and H. Li "Primary ambient extraction for random sign Hilbert filtering decorrelation," in *the International Congress on Acoustics*, Aachen, Germany, 2019.

[6] J. He, *Spatial audio reproduction with primary ambient extraction*, Singapore: Springer Publishing Company, 2016.

[7] H. Yang, C. Shi, and H. Li "Acoustic scene classification using CNN ensembles and primary ambient extraction," in *the* 2019 *IEEE AASP Detection and Classification of Acoustic Scenes and Events*, 2019.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Networks," in arXiv:1406.2661, 2014.

[9] J. Y. Zhu, T. Park, P. Isola, A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in arXiv:1703.10593, 2018.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in arXiv: 1710.09412, 2017.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” In *the 26th Annual Conference on Neural Information Processing Systems*, 2012.

[12] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition,”, in arXiv:1512.03385, 2015.