ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE DECISION SCHEMES

Technical Report

Helin Wang[1,†], *Dading Chong*[1,†], *Yuexian Zou*[1,2,*]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China

ABSTRACT

This technical report describes the ADSPLAB team's submission for Task1 of DCASE2020 challenge. Our acoustic scene classification (ASC) system is based on the convolutional neural networks (CNN). Multiple decision schemes are proposed in our system, including the decision schemes in multiple representations, multiple frequency bands, and multiple temporal frames. The final system is the fusion of models with multiple decision schemes and models pre-trained on AudioSet. The experimental results show that our system could achieve the accuracy of 84.5% (official baseline: 54.1%) and 92.1% (official baseline: 87.3%) on the officially provided fold 1 evaluation dataset of Task1A and Task1B, respectively.

***Index Terms*—** Acoustic scene classification, convolutional neural networks, multiple decision schemes

1. INTRODUCTION

Acoustic scene classification (ASC) aims to classify audio as one of a set of categories such as home, street, and office [1]. Detection and Classification of Acoustic Scenes and Events (DCASE) challenges organized by IEEE Audio and Signal Processing (AASP) Technical Committee are one of the biggest competitions for ASC task [2]. The large-scale dataset provided by DCASE2020 [3] presents a challenge for the system's generalization and low complexity.

The report describes the details of ADSPLAB team's submission for Task1A and Task1B of DCASE2020. More specifically, multiple decision schemes and external data improve the system's performance. Based on the convolutional neural networks (CNN), log-Mel spectrogram (Log-Mel), constant-Q transform (CQT), Gammatone spectrograms (Gamma) and Mel Frequency Cepstral Coefficients (MFCC) are used as the input to the networks. Four independent models with different representations are trained and the decision is made by the average voting strategy, which is called the decision scheme in multiple representations (DCMR). Multiple models are trained on different frequency bands respectively and then ensembled, which is the decision scheme in multiple frequency bands (DCMF). In addition, the decision scheme in multiple temporal frames (DCMT) is operated on each frame of the final feature maps by CNN. For Task1A, external data (*i.e.* AudioSet [4]) and all the multiple decision schemes are applied. While for Task1B, the decision schemes in multiple representations and multiple temporal frames are applied. Under the official fold 1 evaluation setup, our system could achieve 84.5% accuracy with 0.611 log loss in the Task1A evaluation set, and 92.1% accuracy with 0.312 log loss in the Task1B evaluation set.

† Helin Wang and Dading Chong contributed equally to this work.
* Yuexian Zou is the corresponding author.

The remainder of this report is organized as follows. Section 2 describes the proposed multiple decision schemes. Section 3 details the architectures of our networks. Section 4 and Section 5 present the details of experiments and results. Section 6 concludes our work.

2. MULTIPLE DECISION SCHEMES

In this section, the conventional CNN-based method and our proposed multiple decision schemes are introduced, which are the decision scheme in multiple representations (DCMR), the decision scheme in multiple frequency bands (DCMF), and the decision scheme in multiple temporal frames (DCMT).

2.1. Conventional CNN-based Method

CNN-based methods were widely used in ASC task, and provided the state-of-the-art performance [5, 6]. To be specific, given an audio clip, 2-d time-frequency representation (*e.g.* Log-Mel) is first extracted. Convolutional layers are then applied to the time-frequency representation $\boldsymbol{M} \in \mathbb{R}^{T \times F}$ to obtain the deep representation $\boldsymbol{M}^{'} \in \mathbb{R}^{t \times f}$.

$$\boldsymbol{M}^{'} = f_{\mathrm{cnn}}\left(\boldsymbol{M}; \theta_{\mathrm{cnn}}\right) \tag{1}$$

Here, f_{cnn} denotes the operation of the convolutional layers and θ_{cnn} denotes the model parameters of the convolutional layers. The global pooling layer and fully-connected layers are then applied to obtain the predicted score of classification. Let f_{gp}, f_{fc} be the operations of the global pooling layer and the fully-connected layers, respectively. The predicted score $\hat{\boldsymbol{y}} \in \mathbb{R}^{N}$ (where N denotes the number of categories) can be obtained by

$$\hat{\boldsymbol{y}} = f_{\mathrm{fc}}\left(f_{\mathrm{gp}}\left(\boldsymbol{M}^{'}\right); \theta_{\mathrm{fc}}\right) \tag{2}$$

where θ_{fc} denotes the model parameters of the fully-connected layers.

2.2. Decision Scheme in Multiple Representations

Instead of inputing single representation to the networks, multiple representations are used in our system (*i.e.* Log-Mel, CQT, Gamma, and MFCC). One intuitive approach [7] to apply multiple representations is inputting the multi-channel feature maps $\boldsymbol{M}^{*} \in \mathbb{R}^{n \times T \times F}$, where n denotes the number of representations. However, different representations have different characteristics and a single CNN network cannot model the differences effectively. In addition, the same regions in different feature maps reflect different

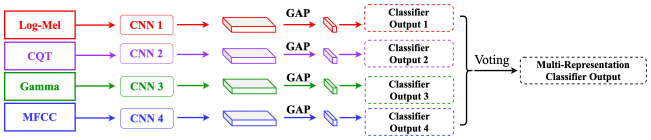

Figure 1: The illustration of DCMR.

frequency information, which may cause the mismatched problem. To overcome these problems and make better use of the representations, we train several independent CNN models based on different representations. As shown in Figure 1, the predicted scores of the models are then summed up to obtain the final predicted score, which is also known as the average voting strategy.

$$\hat{\boldsymbol{y}} = \frac{1}{4}\left(\hat{\boldsymbol{y}}_1 + \hat{\boldsymbol{y}}_2 + \hat{\boldsymbol{y}}_3 + \hat{\boldsymbol{y}}_4\right) \tag{3}$$

where $\hat{\boldsymbol{y}}_1$, $\hat{\boldsymbol{y}}_2$, $\hat{\boldsymbol{y}}_3$, $\hat{\boldsymbol{y}}_4$ denotes the predicted score of CNN model with the input representation of Log-Mel, CQT, Gamma, and MFCC, respectively.

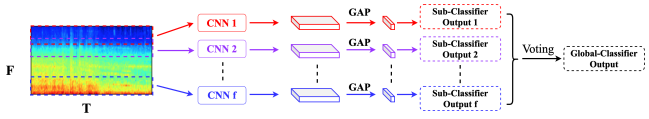

Figure 2: The illustration of DCMF.

2.3. Decision Scheme in Multiple Frequency Bands

The spatial regions of the feature maps are treated equally in the conventional CNN-based methods, however, different acoustic scenes show different activity on the frequency bands [8]. Therefore, we take the sub-spectrograms [8] as input and train several classifiers. Different from [8], the final decision is made by the average voting strategy rather than training a global classifier, which shows better performance in our experiments. As shown in Figure 2, for f sub-spectrograms, the final score is obtained by

$$\hat{\boldsymbol{y}} = \frac{1}{f}\sum_{i=1}^{f}\hat{\boldsymbol{y}}_i \tag{4}$$

2.4. Decision Scheme in Multiple Temporal Frames

Several temporal devisions have been studied in [9] for ASC task, including non division, non-overlap division and overlap division. Among them, overlap division shows the best performance. In this work, a decision scheme in multiple temporal frames is proposed, which feeds the whole audio clip to the network and makes decision on each temporal frame after CNN. Thus, the decision made by each frame could take into account the information of neighboring frames. As shown in Figure 3, for the final feature map $\boldsymbol{M}^{'} \in \mathbb{R}^{t\times f}$, global pooling is applied to the frequency bands and the classifier is then applied to each temporal frames.

$$\hat{\boldsymbol{y}} = \frac{1}{t}\sum_{i=1}^{t}\hat{\boldsymbol{y}}_i \tag{5}$$

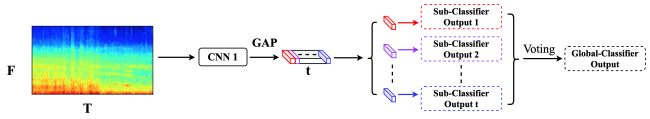

Figure 3: The illustration of DCMT.

3. NETWORK ARCHITECTURES

Our base network architectures are shown in Table 1. Task1A network is a VGG [10] style network, similar to [11]. Batch normalization and Rectified Linear Units (ReLU) are used following the convolutional operations. Global pooling is applied after the last convolutional layer to obtain fixed-length vectors, which is operated by global average pooling in the frequency axis and global max pooling in the temporal axis [11]. Two fully-connected layers followed with a softmax function are then applied to obtain the prediction for classification. Dropout with a ratio of 0.5 is applied between the fully-connected layers. While for Task1B, a tiny CNN is employed to achieve the low complexity, and other setups are the same as Task1A.

Table 1: Network Architectures

Task1A	Task1B
Conv 3×3 @ 64, BN, ReLU Conv 3×3 @ 64, BN, ReLU	Conv 7×7 @ 32, BN, ReLU
Avg Pooling 4×2	Avg Pooling 4×2
Conv 3×3 @ 128, BN, ReLU Conv 3×3 @ 128, BN, ReLU	Conv 7×7 @ 32, BN, ReLU
Avg Pooling 4×2	Avg Pooling 4×2
Conv 3×3 @ 256, BN, ReLU Conv 3×3 @ 256, BN, ReLU	Conv 3×3 @ 64, BN, ReLU
Avg Pooling 2×2	Avg Pooling 2×2
Conv 3×3 @ 512, BN, ReLU Conv 3×3 @ 512, BN, ReLU	Conv 3×3 @ 64, BN, ReLU
Global Pooling	Global Pooling
FC 512, ReLU	FC 200, ReLU
FC 10, softmax	FC 3, softmax

4. EXPERIMENTS ON TASK1A

4.1. Experimental Setups

For Task1A, all the raw audios are resampled to 44.1kHz and fixed to the certain length of 10s by zero-padding or truncating. Log-Mel, CQT, Gamma, and MFCC are then extracted with window size 2048 (46ms) and hop length 512 (11.6ms). The number of frequency bands are 40, 64, 64 and 40, respectively.

In the training phase, the Adam algorithm [12] is employed as the optimizer with the default parameters. The model is trained end-to-end with the initial learning rate of 0.001 and the exponential decay rate of 0.91 for each 200 iterations. Parameters of the networks are learned using the categorical cross entropy loss. Batch size is set to 64 and training is terminated after 12000 iterations. Data augmentation methods Mixup [13] is applied in our experiments to prevent the system from over-fitting and improve the performance.

Table 2: Comparison of accuracy and log loss on Task1A evaluation dataset

Model	Accuracy	Log loss
DCASE2019 Task1A Baseline	46.5%	1.578
DCASE2020 Task1A Baseline	54.1%	1.365
Log-Mel CNN	72.1%	0.879
CQT CNN	71.2%	0.870
Gamma CNN	76.1%	0.762
MFCC CNN	63.6%	1.029
DCMR	79.4%	0.696
Log-Mel CNN + DCMF	75.5%	1.135
CQT CNN + DCMF	74.5%	1.185
Gamma CNN + DCMF	78.8%	1.169
MFCC CNN + DCMF	60.9%	1.801
DCMR + DCMF	80.9%	0.737
Log-Mel CNN + DCMT	74.5%	0.987
CQT CNN + DCMT	73.3%	1.032
Gamma CNN + DCMT	78.2%	0.866
MFCC CNN + DCMT	67.6%	1.081
DCMR + DCMT	79.1%	0.701
CNN10	76.1%	0.634
CNN14	78.5%	0.620
ResNet38	80.3%	0.601
Wavegram-CNN	74.2%	0.830
Wavegram-Logmel-CNN	80.3%	0.606
Ensembled	82.4%	0.553
DCMR + DCMF + DCMT	81.8%	0.694
DCMR + Ensembled	84.2%	**0.569**
DCMR + DCMF + DCMT + Ensembled	**84.5%**	0.611

4.2. Experimental Results

Apart from the models with DCMR, DCMF and DCMT, AudioSet [4] is used as the external data in our experiments. We pretrain models of CNN10, CNN14, ResNet38, Wavegram-CNN and Wavegram-Logmel-CNN with the audio tagging task [14] on AudioSet, and then finetune these models in the ASC task. The ensembled model of all the pretrained models on AudioSet is called Ensembled. Table 2 demonstrates the test results of different models. Among them, DCMR + DCMF + DCMT + Ensembled achieves the highest accuracy, which shows that our proposed DCMR, DCMF, DCMT and using external data can improve the performance for ASC. However, DCMR + DCMF + DCMT + Ensembled performs worse than DCMR + Ensembled on the metric of log loss, which is because more ensembled models are employed in DCMR + DCMF + DCMT + Ensembled and the predicted scores become more smooth.

5. EXPERIMENTS ON TASK1B

5.1. Experimental Setups

In order to achieve the low complexity, multi-channel feature maps (MC) are used for Task1B instead of DCMR. In addition, DCMT is applied because of no extra parameters. The overall architecture is shown in Figure 4. Three representations are used, *i.e.* Log-Mel, CQT and Gamma, and the number of frequency bands are all 64. Other experimental setups are the same as Task1A.

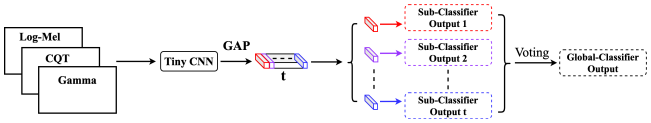

Figure 4: The illustration of Task1B network.

Table 3: Comparison of accuracy and log loss on Task1B evaluation dataset

Model	Accuracy	Log loss	Model size
DCASE2020 Task1B Baseline	87.3%	0.437	450 KB
Log-Mel CNN	87.5%	0.428	468 KB
Log-Mel CNN + DCMT	90.8%	0.371	468 KB
MC + DCMT	**92.1%**	**0.312**	491 KB

5.2. Experimental Results

As presented in Table 3, our model outperforms the official baseline with the similar model size. There is no extra data used in Task1B and we donot use any teacher-student strategy, which obviously shows the effectiveness of MC and our proposed DCMT.

6. CONCLUSION

In this technical report, we detailed our systems to tackle Task1A and Task1B of the DCASE2020 challenge. Multiple decision schemes (*i.e.* DCMR, DCMF, and DCMT) have been proposed and greatly improved the performance for ASC task. These schemes were designed to fit the audio characteristics, and we believe they can offer good generalization properties for other audio processing tasks.

7. ACKNOWLEDGMENT

This work was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: JCYJ20170817160058246 & JCYJ20180507182908274). Thanks for the code[1] provided by Qiuqiang Kong.

8. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, "Computational analysis of sound scenes and events," 2017.

[2] http://dcase.community/challenge2020/.

[3] T. Heittola, A. Mesaros, and T. 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. [Online]. Available: https://arxiv.org/abs/2005.14623

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[1] https://github.com/qiuqiangkong/dcase2019_task1

[5] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," *arXiv preprint arXiv:1907.06639*, 2019.

[6] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Cp-jku submissions to dcase'19: Acoustic scene classification and audio tagging with receptive-field-regularized cnns."

[7] D. Ngo, H. Hoang, A. Nguyen, T. Ly, and L. Pham, "Sound context classification basing on join learning model and multi-spectrogram features," *arXiv preprint arXiv:2005.12779*, 2020.

[8] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 825–829.

[9] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," *DCASE2018 Challenge*, 2018.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[14] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *arXiv preprint arXiv:1912.10211*, 2019.