ACOUSTIC SCENE CLASSIFICATION WITH DECOMPOSED CONVOLUTION NEURAL NETWORKS

Technical Report

Minhan Kim[1], SeungHyeon Shin[1], Seungjae Baek[1], Seokjin Lee[1,2], Sooyoung Park[3], Youngho Jeong[3],

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu, Republic of Korea, {kmh7576, sh.shin, baek7350, sjlee6}@knu.ac.kr
[2] School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea
[3] Media Coding Research Section, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea, {sooyoung, yhcheong}@etri.re.kr

ABSTRACT

This report describes a model submitted to DCASE2021 Task 1 subtask A. Our model is developed by applying canonical polyadic decomposition to the conventional convolutional-neural-network-based models to reduce the model size to achieve the goal of Task 1A. More specifically, we apply the decomposition method to dual ResNet, which divides the features into two parts along the frequency axis and processes them independently, and shallow inception model. In order to evaluate our model, a simulation for acoustic scene classification was performed with the development dataset of DCASE 2021 Task 1A, and our model showed about log loss of 1.03-1.06 and macro accuracy of 62%-66% far better than that of the baseline model. Also, the model size of our system is smaller than 128 kbytes, which is the limit of the DCASE2021 Task 1A.

Index Terms— Acoustic scene classfication, Resnet, Shallow inception, Mean-teacher, Decomposed convolution, Model compression

1. INTRODUCTION

In order to provide services for various persons with machines, it is requied to understand the environments and situations. Recently, many machine learning algorithms have been developed to provide personally optimized services with analysis of multiple sensor signals. The acoustic scene classification (ASC) task is one of the problems to understand environment with sound signals, and some competitions have been held to solve the ASC problem, such as DCASE 2021 Task 1[1].

Undoubtedly, great progress has been made in research to increase the accuracy of machine learning techniques. In order to apply the algorithms to real machnies successfully, low memory and computational power have to be considered. The DCASE 2021 Task 1A [2] focuses on this problem, and the main goal of the task is that the model must have a minimal computational complexity while maintaining sufficiently high performance.

In order to achieve the goals of the task, we focus on the model structure optimization and compression. More specifically, we reduce the model size by using the decomposed convolution layers instead of the conventional convolution layers, which is a major part of the model parameters.

2. PROPOSED METHODS

2.1. Model compression

The previous studies[3, 4] have shown that Canonical Polyadic(CP) decomposition can effectively compress convolutional layers. Figure 1 shows how CP decomposition reconstructs a convolutional layer. The CP-decomposed convolution layer consists of three

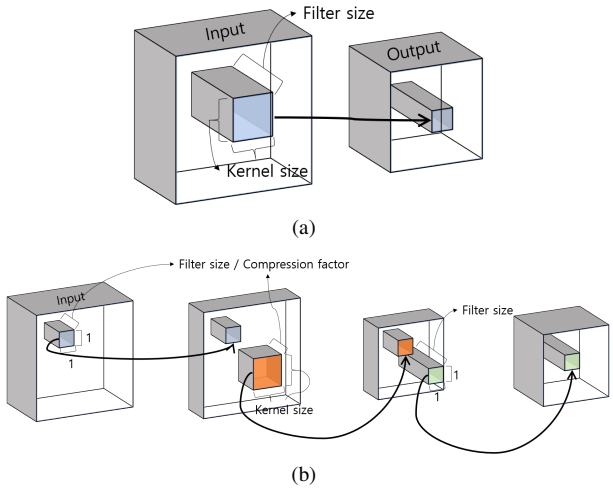

Figure 1: (a) Convolution layer (b) Decomposed convolution layer

CNN layers. The first CNN layer has $\lfloor C_{in}/K \rfloor$ output channles with (1×1) kernels, where C_{in} and K are number of input channels and compression factor, respectively, and $\lfloor \rfloor$ means a floor operator. The second layer has $\lfloor C_{in}/K \rfloor$ input channels and $\lfloor C_{in}/K \rfloor$ output channels with the same kernel size as the original convolution layer. The third layer has C_{out} output channels with (1×1) kernels, where C_{out} is the number of output channels of the original convolution layer. The decomposed convolution layer can reduce $(C_{in} \times C_{out} \times N_{kernel})$ parameters to $\left[C_{in} \times \lfloor C_{in}/K \rfloor + (\lfloor C_{in}/K \rfloor)^2 \times N_{kernel} + \lfloor C_{in}/K \rfloor \times C_{out}\right]$ parameters, where N_{kernel} is the kernel size.

To further reduce the size of the trained model, we quantized the weights of the model to 16 bits.

2.2. Model architectures

2.2.1. *Dual ResNet model*

The frequency-aware parallel structure was inspired by [5, 6]. The *Trident* parallel structure in [6] divides the input into 3 parts along the frequency axis and feeds each frame into ResNet. Each ResNet learns the characteristics of each segmented frame. Our *Dual* model consists of two paths of 0-128 and 128-256 mel bins. We have checked through experiments that the two paths structure works well enough. The residual block of our model consist of decomposed convolution layers with kernel size 3×3 and compression factor 4. And to further compress the model, we used parameter sharing in the CP decomposed convolutional layer. After every convolution block, we added skip connection and last Max pooling layer optional for each block as shown in Figure 2.

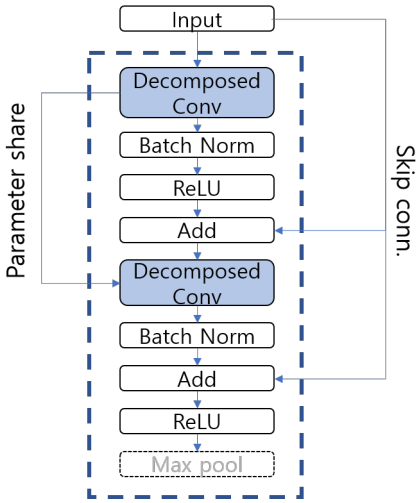

Figure 2: Residual block with parameter sharing and skip connection.

The ResNet structure composed of this residual block can be seen in Figure 3. Here, we add a bottleneck structure to the last

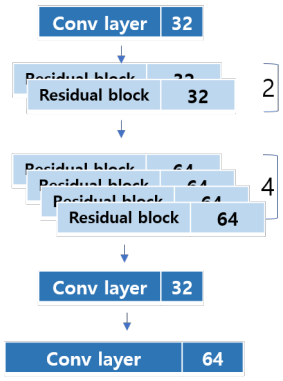

Figure 3: ResNet blocks with filter size.

stage to control the number of model parameters. The output of each ResNet is concatenated and output through a global max pooling and softmax activation layer. The overall structure of model is shown in Figure 4.

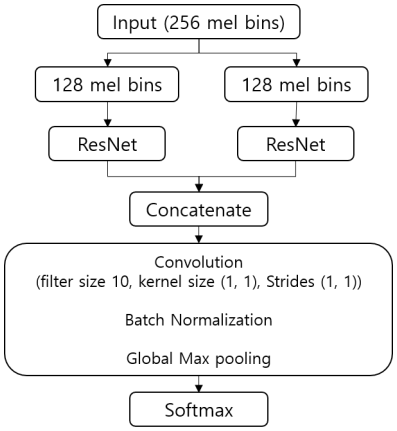

Figure 4: Overall structure of Dual ResNet model.

2.2.2. *Shallow inception model*

A previous study[6] demonstrated in DCASE 2020 task 1B that proposed shallow inception model is effective for light models. We modified this model to fit the model complexity constraints. The overall model architecture of shallow inception is shown in Figure 5.

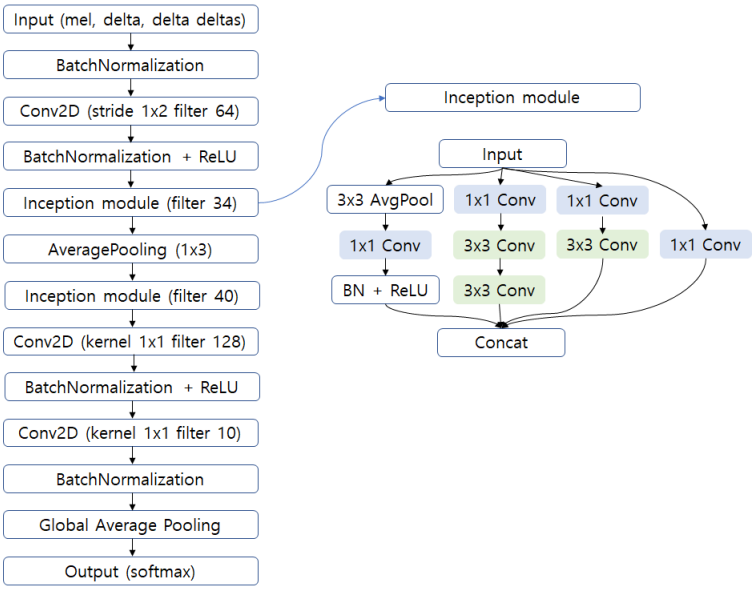

Figure 5: Overall structure of Shallow inception model

Since the model complexity depends on the filter size of the convolution layer, we first reduced the filter size of each inception module without compromising performance. And the inception module shares 1×1 and 3×3 convolution layers in each module.

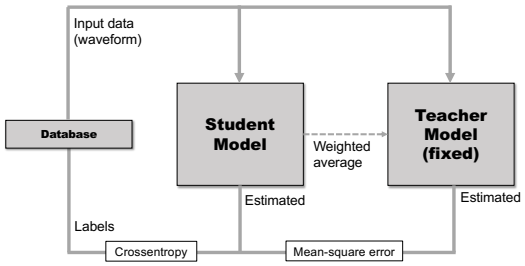

Figure 6: A block diagram of the mean-teacher model

2.2.3. Mean teacher model

In order to enhance the performance, ensembling a number of models may be a good strategy sometimes. Unfortunately, the ensembling strategy cannot be applied to solve task 1A because it requires a large number of parameters. Instead of applying model ensembling directly, we comprise a mean-teacher model [7] to take advantage of the ensembling. The mean-teacher model consists of the teacher and student models as shown in Figure 6. The parameters of the teacher model are not trained during the training process, and they are updated at the end of each iteration as

$$\theta_{teacher} \leftarrow (1-\alpha)\,\theta_{teacher} + \alpha\theta_{student} \quad (1)$$

where $\theta_{teacher}$ and $\theta_{student}$ are parameters of teacher and student models, respectively, and α is an average coefficient for exponentially tapered moving average. The loss function is set to weighted sum of two losses as

$$C_{total}\left(\theta_{student}\right) = C_{class}\left(\theta_{student}\right) + \beta C_{consist}\left(\theta_{student}\right) \quad (2)$$

where C_{class} is a classification cost between the prediction results of the student model and the ground truth and $C_{consist}$ is a consistency cost between the prediction results of the student and teacher models. In our system, the classification cost is set to the same as the Dual ResNet without applying the mean-teacher model, and the consistency cost is set to mean-squared error between two model outputs.

3. EXPERIMENTAL SETTINGS

3.1. Data Preprocessing

The data of DCASE 2021 Task 1 subtask A[8] is 10 second long 48kHz sampled audio file. We loaded it at a 44.1 kHz sample rate and passed it through a filter with a filter length of 2048 and a hop size of 1024 and converted to a spectrogram. And each spectrum was compressed through a Mel filter with a number of bins of 256, and the log was taken to create a log Mel spectrogram. Lastly, deltas and delta-deltas were calculated from log Mel spectrogram and concatenated to the channel axis.

3.2. Data Augmentation

A mixup[9] was used during each mini-batch to generalize the model in training. We set the mixup parameter alpha to 2.0 for Dual ResNet and 0.2 for shallow inception.

3.3. Training Setup

We used categorical cross-entropy as the loss function, and for the optimizer, we used SGD with 0.9 momentum in the dual resnet and focal loss[10] in the shallow inception. The learning rate lr was modulated using a cosine annealing learning rate scheduler with restart to avoid local minima and find a deeper optimal point. The restart epochs was set to 2, 6, 14, 32, 60, 100, 130, 180, 210, 220, 250, 270, 290, 310 and 340. The initial lr was set to 0.5 in the dual ResNet model and 0.01 in the shallow inception model. And lr was reduced to 10^{-4}. With each restart, restart lr decreased by 10 %.

4. RESULTS

To test each model architecture, training and test were performed using the train/test split provided by the competition. Table 1 shows a performance comparison for each cnn structure. DuRes-MT means the Dual ResNet model trained by using the mean-teacher structure.

Table 1: Results of development set

ID	system name	log loss	accuracy	model size
1	Dual ResNet	1.068	64%	125.7 KB
2	Shallow Inception	1.040	62%	125.1 KB
3	DuRes-MT (student)	1.047	67%	125.7 KB
4	DuRes-MT (teacher)	1.035	65%	125.7 KB
	Baseline	1.473	47%	-

The result shows that the developed systems have log loss performance of 1.035 - 1.068 and accuracy performance of 62 % - 67%. All systems have much better performance than the baseline. The shallow inception model have better log loss but worse accuracy performances than the dual ResNet model. The mean-teacher structure seems to enhance the performance of dual ResNet model. The optimal performance of the teacher model was better than that of the student model, but it was not consistent throughout our experiment.

5. ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00050, Development of Human Enhancement Technology for auditory and muscle support).

6. REFERENCES

[1] http://dcase.community/challenge2021/.

[2] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems," *arXiv preprint arXiv:2105.13734*, 2021.

[3] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "Cp-jku submissions to dcase'20: Low-complexity cross-device acoustic scene classification with rf-regularized cnns," DCASE2020 Challenge, Tech. Rep, Tech. Rep., 2020.

[4] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," *arXiv preprint arXiv:1412.6553*, 2014.

[5] W. Gao and M. McDonnell, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," *Tech. Rep., DCASE2019 Challenge*, 2019.

[6] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with cnn variants," DCASE2020 Challenge, Tech. Rep, Tech. Rep., 2020.

[7] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[8] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," *arXiv preprint arXiv:2005.14623*, 2020.

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

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.