KT SUBMISSION: PERIODIC ACTIVATION AND KNOWLEDGE DISTILLATION FOR DATA-EFFICIENT LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION

Technical Report

TaeSoo Kim, Jiwon Kim, Daniel Rho, Gahui Lee, JaeHan Park

KT, South Korea

ABSTRACT

This technical report describes our approach to participate in DCASE 2024 Challenge Task 1: Data-Efficient Low-Complexity Acoustic Scene Classification. Our main contribution to this work is the use of an improved backbone model, a modified BC-Res2Net with the GhostNet module. In addition, to improve generalization performance even with a limited amount of training data, we adopted the Snake activation function, which is known to be robust to unseen data due to its extrapolation capabilities. Through experiments, we demonstrated that our model significantly improves acoustic scene classification performance, especially when the number of training samples is limited.

Index Terms— Acoustic Scene Classification

1. INTRODUCTION

Detection and Classification of Acoustic Scenes and Events (DCASE) is an annual audio-related challenge that aims to explore a variety of audio-related tasks, such as acoustic scene classification (ASC), audio captioning, audio retrieval, and audio-based traffic monitoring. Among them, we focus on task 1: Data-Efficient Low-Complexity ASC as the primary research.

As it has been, the goal of this task is to achieve higher efficiency in ASC. Similar to previous years, limitations on model size and the number of multiply-accumulate operations (MACs) remain consistent to address constrained computational resources. In addition to computational efficiency, one more aspect of efficiency was introduced this year: data efficiency. Participants also have to achieve data efficiency in achieving robust acoustic scene classification. To this end, they were given five different subsets with different sample sizes, spanning from 5% to 100% of the original training dataset, to address the scarcity of labeled data applicable to real-world contexts.

In this report, we mainly investigate the BCRes2Net [1, 2] with the GhostNet V2 module [3, 4], aiming to maintain high classification accuracy under limited data conditions. Additionally, we utilized the Snake function [5] as an activation function instead of a Swish function inside BCRes2Net. Furthermore, we inspect the knowledge distillation method using two pre-trained models, PaSST [6] and EAT [7], to reinforce the proposed model. With our model, we achieved about 6.5% accuracy improvement on average.

The remainder of the report is organized as follows: Section 2 explains a detailed exposition of the proposed systems, showing their robust performance recognition, particularly when faced with constraints on data availability. Section 3 discusses the results of our experiments including experimental settings and evaluation results. Section 4 presents our conclusion.

2. PROPOSED SYSTEM

2.1. Model Architecture

The basic structure of our proposed model is based on BCRes2Net[1, 2]. BCRes2Net utilizes both 1D convolution and 2D convolution, combining frequency depth-wise convolution and temporal depth-wise convolution operations to perform residual operations. The model with its small number of parameters extracts feature maps along the frequency and time axes respectively and combines them effectively. It shows high performance in both keyword spotting[1] and ASC tasks [2, 8, 9].

In our previous study [9], we applied the GhostNet V2 [3, 4] module to BCRes2Net. The GhostNet V2 module is designed to enhance the efficiency of feature map extraction by limiting the extraction of redundant features. This is achieved by utilizing a series of low-cost operations that generate more features from intrinsic features through a lightweight and cost-effective structure. Additionally, the decoupled fully connected attention (DFC Attn) module is incorporated which enhances the intermediate features by aggregating local and long-range information simultaneously, leading to improved performance.

2.2. Periodic Activation Function

BCRes2Net-based networks [1, 2, 9] use the Swish function [10] as an activation function after every temporal depth-wise convolution operation. The Swish function is defined as follows:

$$Swish(x) = x + \sigma(x), \tag{1}$$

where $\sigma(\cdot)$ indicates the sigmoid operation. Although the Swish activation function is known to be effective in various tasks, we sought to replace this activation function with another one to achieve better generalization performance even with a smaller amount of training data.

As an alternative, we chose the Snake function [5] and replaced the Swish with the Snake function. The Snake function can be defined as follows:

$$Snake(x) = x + \frac{sin^2(\alpha x)}{\beta}, \tag{2}$$

where α and β control the periodic part [5]. The main reason for adopting the Snake function is that it is known to be robust to unseen samples due to its periodic component ($sin^2(x)$). That is, the Snake function can be useful, especially when training samples are scarcely given.

Table 1: Specification of Snake-Ghost-BC-Res2Net. F, T, and C denote the size of frequency, time, and channel dimensions, respectively. Also, PQ and FP16 indicate 8-bit post-quantization and 16-bit weight precision, respectively. The values in parentheses for the input column indicate the number of channels when using FP16. "DFC Attn" denotes a decoupled fully connected attention module.

Stage	Input (F×T×C)	Operation	DFC Attn	C	Strides
stem	F × T × 1	Conv + ReLU + BN	-	2C	2
1	F/2 × T/2 × 2C	Ghost-BC-Res2Net w/ Snake act. Block	True	C	-
	F/2 × T/2 × C	Ghost-BC-Res2Net w/ Snake act. Block	False	C	-
2	F/2 × T/2 × C	Max-Pool2d	-	-	2
	F/4 × T/4 × C	Ghost-BC-Res2Net w/ Snake act. Block	True	1.5C	-
	F/4 × T/4 × 1.5C	Ghost-BC-Res2Net w/ Snake act. Block	False	1.5C	-
3	F/4 × T/4 × 1.5C	Max-Pool2d	-	-	2
	F/8 × T/8 × 1.5C	Ghost-BC-Res2Net w/ Snake act. Block	True	2C	-
	F/8 × T/8 × 2C	Ghost-BC-Res2Net w/ Snake act. Block	False	2C	-
	F/8 × T/8 × 2C	ResNorm	-	2C	-
4	F/8 × T/8 × 2C	Ghost-BC-Res2Net w/ Snake act. Block	True	2.5C(PQ)/2C(FP16)	-
	F/8 × T/8 × 2.5C(2C)	Ghost-BC-Res2Net w/ Snake act. Block	False	2.5C(PQ)/2C(FP16)	-
	F/8 × T/8 × 2.5C	Ghost-BC-Res2Net w/ Snake act. Block	False	2.5C(PQ)	-
head	F/8 × T/8 × 2.5C(2C)	Global Avg. Pool	-	-	-
	1 × 1 × 2.5C	Linear	-	num classes	-

Table 2: Test Accuracy (%) for each train subset of each model

				Train subset size					
Architecture	TYPE	FLOPS (MAC)	# params	5	10	25	50	100	average
Baseline		29.42 M	61,148	42.40	45.29	50.90	53.19	56.99	49.75
Ghost-BC-Res2Net	LARGE	28.56 M	85,824	46.56	51.87	56.50	58.57	60.97	54.89
+ w/ Snake Act. (PQ)	LARGE	28.56 M	85,824	**49.87**	52.99	56.00	58.55	61.30	55.74
+ w/ Snake Act. (FP16)	SMALL	26.48 M	63,992	49.72	**53.17**	**58.35**	**59.39**	**61.94**	**56.51**

2.3. Knowledge Distillation

Knowledge distillation [11] is a technique to improve the performance of a model through the distillation of knowledge from the teacher model. Several studies have demonstrated that using knowledge distillation techniques can enhance the performance of models in ASC task [12, 13].

For example, PaSST has been proven to be an effective teacher model for improving classification performance in low-complexity CNNs [13]. As such, we applied a knowledge distillation strategy with two teacher models: PaSST [6] and EAT [7]. EAT achieved state-of-the-art performance on the AudioSet classification task. To introduce diversity into the soft labels generated by the teacher, we applied online distillation to the model, that is, we alternately trained the teacher and student models at each training step.

2.4. Model Compression

To meet the task constraints, we took two different approaches. The first approach is applying Post Quantization (PQ) after training with 32-bit precision weights to have an 8-bit precision model. The other involves using mixed precision, having FP16 precision weights. As using 16-bit precision weights needs twice the memory of that of 8-bit precision weights, we used a smaller network architecture to meet the constraint. More precisely, we removed the last block in the last stage of the model. In addition, we reduced the number of channel multipliers of the last stage from 2.5 to 2 (Tab. 1). As shown in Tab. 2, the model after PQ had approximately 86k parameters with 28.56 MMACs, whereas the model with FP16 precision had 64k parameters with 26.48 MMACs.

3. RESULTS

3.1. Experimental Settings

We built our system on top of the publicly available baseline code [14], which was provided by the host of the challenge. We trained models with TAU Urban Acoustic Scenes 2022 Mobile dataset [15] and did not use any other dataset during training.

Regarding input features, we employed the log mel-spectrogram with 288 mel bins after resampling audio samples to a 32 kHz sampling rate. The window size was set to 3072, the hop size to 500, and the number of FFT points to 4096. We used frequency masking [16], freq-mixstyle [12, 17], and time rolling for data augmentation. In addition, we utilized device impulse responses from the Microphone Impulse Response Project (MicIRP) [18] to augment audio files to train models to be robust to various recording environments, as done by Shmid et al. [13].

Regarding optimization, we used the AdamW along with cosine annealing with a warmup scheduler. The batch size was set to 256, and all proposed models were trained for 200 epochs. For each train subset size, we used the same shared training hyperparameters,

Table 3: Test Accuracy (%) for each train subset of our model using knowledge distillation

Student Model	Teacher Model	Train subset size 5	10	25	50	100	average
Ghost-BC-Res2Net w/ Snake Activation FP16	EAT Base (Fine-tuning on AS-2M) EP30	47.82	53.11	**57.60**	59.21	61.6	55.87
	EAT Base (Fine-tuning on AS-2M) EP10	**49.10**	**53.18**	55.93	59.60	61.84	55.93
	EAT Base (Pre-training) EP 30	48.81	52.05	56.5	58.93	61.77	55.61
	PaSST-S SWA P16 F128 (AP476)	48.09	52.54	56.65	58.79	61.54	55.52
	PaSST-S SWA P16 F128 (AP4761)	48.05	52.14	57.34	**59.79**	**62.55**	**55.97**

including learning rate and batch size. That is, we did not tune training-related hyperparameters for different subset sizes.

For knowledge distillation, we selected Ghost-BC-Res2Net with Snake activation FP16 model as a student model. We adjusted the input data to match the settings used during the pre-training of the teacher model and trained it for 150 epochs. We set the learning rate of the teacher model to be 0.01 times smaller than that of the student model. We use the following knowledge distillation loss function [13]:

$$L = \lambda L_{ce}(\delta(z_S), y) + (1-\lambda)\tau^2 L_{kd}(\delta(z_S/\tau), \delta(z_T/\tau)), \quad (3)$$

where L_{ce} is cross entropy loss for hard labels and L_{kd} is Kullback-Leibler divergence loss for soft labels. δ denotes log-softmax activation function and τ denotes temperature which we set to 2.0. z_S and z_T are logits from student and teacher respectively, and y are hard labels. λ denotes the distillation loss coefficient and is set to 0.02. In this work, we applied online distillation [19], since our teacher model has not been pre-trained on the current task data.

3.2. Experimental results

Tab. 2 provides quantitative results on the TAU validation dataset, each trained with a different training data size. As the table shows, replacing the backbone architecture with our proposed network architecture significantly improves performance regardless of the training data size, from at least 10.3 % to 12 % increase in test accuracy. After applying the Snake activation function, our experiments showed that the model's performance improved compared to the model without the Snake activation function. As a result, the average accuracy over various train subset sizes improved by approximately 3%. This indicates that the Snake activation function enhances the model's extrapolation capabilities, allowing it to operate robustly on unseen signals. The model with PQ demonstrated a notable performance improvement of 7.1% on subset 5 when only a small dataset was allowed. Furthermore, the model with FP16 precision, despite using fewer parameters than the PQ-applied model and Ghost-BC-Res2Net, demonstrated an average performance improvement of 1.3% and 3%, respectively.

Tab. 3 shows the results on the TAU validation set for the model trained using online knowledge distillation with PaSST and EAT as teachers. For the models trained with EAT, the fine-tuned model on AudioSet provides better average accuracy than the pre-trained model. PaSST-S (AP4761) shows the best performance among the models trained with knowledge distillation. Specifically, on subsets 50 and 100, it achieved an accuracy improvement of 0.6% and 0.9%, respectively, compared to the model trained without knowledge distillation. Although the models trained with knowledge distillation showed higher performance in some experiments, the overall average accuracy was degraded.

4. CONCLUSION

In this report, we describe our method for achieving data- and computationally-efficient ASC, using Snake activation functions with Ghost-BC-Res2Net and applying knowledge distillation techniques. The Snake function, which has both the advantage of monotonic and periodic activation functions, demonstrated its effectiveness in classifying acoustic scenes robustly even with a small amount of training data. Through online knowledge distillation, we observed performance improvements in some experiments; however, achieving even better performance may require optimizing the training hyperparameter settings. Our experimental results demonstrate the superiority of our approach, especially in a very data-scarce environment.

5. REFERENCES

[1] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," *arXiv preprint arXiv:2106.04140*, 2021.

[2] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[3] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[4] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "Ghostnetv2: Enhance cheap operation with long-range attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9969–9982, 2022.

[5] L. Ziyin, T. Hartwig, and M. Ueda, "Neural networks fail to learn periodic functions and how to fix it," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1583–1594, 2020.

[6] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*. ISCA, 2022, pp. 2753–2757. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-227

[7] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "Eat: Self-supervised pre-training with efficient audio transformer," *arXiv preprint arXiv:2401.03497*, 2024.

[8] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method

using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE2022 Challenge, Tech. Rep., June 2022.

[9] T. Kim, D. Rho, G. Lee, and J. H. Park, "Dual-strategy enhancement of acoustic scene and event classification: Integrating res2net, ghostnet, and mobileformer architectures," DCASE2023 Challenge, Tech. Rep., May 2023.

[10] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[11] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network." *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1503.html#HintonVD15

[12] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[13] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 161–165.

[14] "DCASE 2024 challenge task 1 baseline," https://github.com/CPJKU/dcase2024_task1_baseline, 2024.

[15] 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, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[16] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[17] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain Generalization with Relaxed Instance Frequency-wise Normalization for Multi-device Acoustic Scene Classification," in *Proc. Interspeech 2022*, 2022, pp. 2393–2397.

[18] "Microphone impulse response project," http://micirp.blogspot.com/, 2024.

[19] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.