CLOVA SUBMISSION FOR THE DCASE 2021 CHALLENGE: ACOUSTIC SCENE CLASSIFICATION USING LIGHT ARCHITECTURES AND DEVICE AUGMENTATION

Technical Report

Hee-Soo Heo[*1], *Jee-weon Jung*[*1], *Hye-jin Shim*[2], *Bong-Jin Lee*[1]

[1]Naver Corporation, South Korea, [2]School of Computer Science, University of Seoul

ABSTRACT

This technical report addresses the submitted system of Naver Clova for the DCASE 2021 challenge task 1-a. The aim is to develop an acoustic scene classification system that can generalize towards unknown devices using a DNN with a limited number of parameters. We propose two lightweight architectures using residual networks, a method referred to as attentive max feature map, and multitask learning. After the initial training, the model is further fine-tuned using knowledge distillation. Two augmentation methods are also explored to simulate various recording devices. The proposed two architectures have 63,547 and 65,424 non-zeros parameters with a 16-bit resolution, both less than 128KB. Following the official protocol of train and test set split from the TAU Urban Acoustic Scenes 2020 Mobile development dataset, each model achieves 70.48% and 69.68% accuracy respectively.

***Index Terms*—** Attentive max feature map, knowledge distillation, low-complexity, deep neural network, acoustic scene classification

1. INTRODUCTION

Acoustic scene classification (ASC) system can present valuable context information for improving diverse audio-related applications. To adopt ASC model in various applications additionally that can run seamless and in real-time, developing lightweight models become a crucial issue. The detection and classification of acoustic scenes and events (DCASE) 2021 competition focuses on low complexity solutions requiring deep neural networks (DNNs) to have less than 128KB, that is, $32,768$ non-zero parameters in single precision floating point format (32-bit) [1]. Additionally, generalization on diverse unknown devices are also required.

To meet these requirements, we explore various methods. First, we design two DNN architectures with half-precision floating-point (16-bit) by quantizing the single-precision model after training. Both models are first trained using the conventional approach with categorical cross-entropy loss. Then, each model is further trained using a knowledge distillation framework to reduce misclassification in confusing pairs of scenes (e.g., airport and shopping_mall) [2–4]. Diverse data augmentation techniques are also utilized while training the models, both widely used approaches (e.g., mix-up [5]) and proposed approaches (see Section 5). Also, because ensemble becomes impossible for low-complexity solutions, we explore stochastic weight averaging technique which can be utilized to improve the performance without increasing the complexity of a model [6]. Through these methods, our two proposed architectures demonstrate a classification accuracy of 70.48% and 69.68% on the official cross validation setup of the DCASE 2021 task 1-a, respectively.

2. ACOUSTIC FEATURE

We use 128-dimensional mel-spectrograms driven from $1,024$ FFT bins as the input feature for all experiments throughout this report. We adopt delta and delta-delta coefficients following state-of-the-art systems in ASC [7, 8]. We explore two options to compose the input feature. First, we concatenate on the channel dimension making the input shape to $(3, t, f)$ where t and f are the number of time sequences and mel-spectrogram bins, respectively. Second, we concatenate on the frequency dimension following the majority of preceding studies, making the input shape to $(1, t, f \times 3)$.

Pre-emphasis is applied to all raw waveforms before mel-spectrogram extraction. A window size of 40ms and a shift size of 20ms is used, with hamming window function. In the training phase, all audio segments are cropped into a fixed duration. In the evaluation phase, we first compose multiple segments that have the same duration with train phase by shifting the input audio segment. Then, we derive the final output by averaging the model's output on multiple segments.

3. MODEL ARCHITECTURE

We address two lightweight DNN architectures that match low complexity requirement without the use of parameter pruning or int8 resolution quantization. We adopt a half-precision floating-point format for both models allowing up to $65,536$ weight parameters.

3.1. Attentive max feature map

Our first model is designed using a method referred to as attentive max feature map (AMFM) [9]. The AMFM is built on top of MFM that has proven its effectiveness in a number of tasks, especially where relatively small dataset exists (e.g., ASC and audio spoofing detection). Similar to the competitive manner in MFM, AMFM compares the feature maps before and after the attention to alleviate the excessive deletion of information in the conventional attention-based mechanism.

The architecture of our AMFM model for the DCASE 2021 challenge is identical to that of Shim *et al.*, except two adaptations we made to meet low complexity requirement. First, we adopt smaller number of filters, 32 at most. Second, we use a global average pooling instead of a fully-connected layer after the last AMFM block. Following [9], we also allocate additional label to each audio recording as "indoor", "outdoor", and "transportation" and apply

[*] Equal contribution

extended multi-task learning architecture using these labels. This model has $77,572$ parameters including the extended MTL. But, MTL related parameters are removed after training is complete, resulting in $63,547$ parameters.

3.2. ResNet

We design the other model using a variant of ResNetSE architecture [10, 11]. Decomposed convolution layer is adopted that takes up less number of parameters compared to the original convolution layer. For the first convolution layer and the first residual block, however, we use the original version without decomposition based on our empirical results. Also, we replaced the rectified linear unit activation functions in the original ResNetSE with MFM operation [12].

We build four residual blocks constructed with only 17 convolutional layers to meet the limited number of parameters. From the first block to the last block, each block outputs 16, 24, 28 and 32-dimensional feature-map, respectively. Unlike the AMFM model, we do not apply extended MTL on this model, instead we apply various data augmentation methods further addressed in Section 5. This model has $63,547$ parameters.

4. KNOWLEDGE DISTILLATION

Knowledge distillation (KD) has been shown effective for the ASC task where it diminishes the number of mis-classification in confusing pairs of scenes (e.g., airport-shopping_mall and metro-metro_station) [2, 13]. For both AMFM and ResNet models, we apply the KD approach and further improve the performance. The KD scheme that we use mostly follows the recipe used in [3].

After performing initial training, we initialize both teacher and student DNNs using identical weight parameters to perform KD. We use three audio segments from an identical scene with different locations and extract soft-labels from a teacher DNN. Then, the element-wise average of three soft-labels is used to guide the learning of a student DNN. Identical to [2] we use the summation of three loss functions to train the student DNN: cosine distance between last hidden layers, Kullback-Leibler divergence between output layers, and a categorical cross-entropy using the ground truth label. One thing different from [3] is that we update the teacher DNN using the weights of a student DNN whenever a new best accuracy on the fold1 test dataset is achieved. In addition, we do not exploit specialist DNN for KD because we assumed that it would be rather harmful to try distilling multiple DNNs into a light student DNN with less than $100k$ weight parameters.

5. AUGMENTATION

Generalization towards audio segments recorded via an unknown device has recently become an important measure of an ASC model. Because of the limited number of parameters, we assumed that the impact of unknown device would become even more tremendous. To account for unknown devices that exist in the evaluation dataset, we apply various existing data augmentation methods.

Like the majority of recent ASC studies, we first apply mix-up for all our models [5]. At the raw waveform level, tempo change and channel corruption are applied. The tempo change makes the audio speed 0.75 or 0.85 times slower or 1.15 or 1.25 times faster using the sox library. To train a model that generalizes well for various channels, we encode and decode audio signals using MP3 and acc codec.

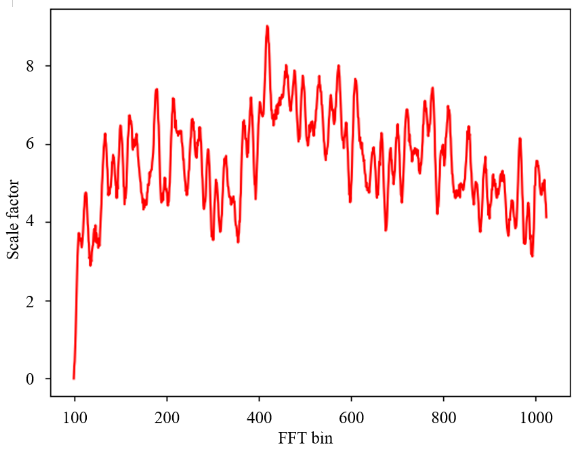

Figure 1: An example of random scale factors on FFT bins.

We also propose and adopt an augmentation technique that randomly scales the amplitude of FFT bins before applying mel-scale filterbanks. We expect that the frequency response of various devices could be simulated by arbitrarily adjusting the amplitude of each FFT bin. Fig. 1 shows an example of random scales on $1,024$ FFT bins. These values are then multiplied to each FFT bin of a spectrogram. This is inspired by spectrum correction technique, that scales FFT bin using known device information and aim to simulate various recording devices [14].

6. EXPERIMENTS

6.1. Dataset

We use the TAU Urban Acoustic Scenes 2020 Mobile development dataset for all experiments and follow the official fold 1 train/test split except two submitted systems that is trained using the entire development set. The training data includes approximately 38 hours of audio recordings and the fold 1 test data includes approximately 8 hours of audio recordings. This dataset has been recorded using three different devices, and include six additional simulated devices. Other details regarding the dataset can be found in [1].

6.2. Configurations

We use the PyTorch library written in Python for all experiments of this report. We use Adam [15] with an initial learning rate of 0.001. A cosine annealing learning rate scheduler is adopted. The number of epochs for initial and KD training is 400 and 130, respectively. For spectrum augmentation, we use α of 0.5. In the training phase, we compose a mini-batch of size 24, with 5s duration for each audio recording. To derive soft-labels used in KD training, we first extract three soft-labels from an identical scene with random locations and average them following [2].

6.3. Result analysis

Table 1 describes the result of four models: AMFM and ResNetSE after initial and KD training. Each column shows device-wise accuracy where "All" means overall accuracy. PyTorch library's number

Table 1: Results of the AMFM and ResNetSE model after initial and KD training is complete. Reported in accuracy (%). Best accuracy per each device is denoted in boldface.

Model	# params	A	B&C	S1∼S3	S4∼S6	All
Official baseline	46,233	-	-	-	-	47.70
AMFM-initial	65,424	69.39	68.39	69.39	66.77	68.30
ResNetSE-initial	63,547	73.93	68.84	67.87	66.96	68.46
AMFM-KD	65,424	72.42	68.39	**70.81**	68.48	69.68
ResNetSE-KD	63,547	**76.06**	**70.51**	69.89	**69.19**	**70.48**

of parameter calculation was used to report the number of parameters for each model where we report the number of non-zero parameters.

All four models of this report outperformed the official baseline [16] with a large margin. For both AMFM and ResNetSE models, KD training demonstrated its effectiveness. One thing worth noting is that by analyzing per device accuracies, KD training improved performance for both known and unknown (devices S4∼S6) devices. In general, ResNetSE outperformed AMFM model slightly where the ResNetSE model after KD training demonstrated an overall accuracy of 70.48%.

As four submissions are allowed for the DCASE 2021 challenge, we comprise our four DCASE challenge submitted systems as follows:

1. Clova_AMFM: AMFM model w/ KD, trained on fold1 configuration
2. Clova_ResNet: ResNetSE model w/ KD, trained on fold1 configuration
3. Clova_AMFM_Whole: AMFM model w/ KD, trained using all $23,040$ audio segments
4. Clova_ResNet_Whole: ResNetSE model w/ KD, trained using all $23,040$ audio segments

All four submitted models adopt KD-based further training. For the first and the second systems, we use the model that performs best on the fold 1 test set, corresponding to the third and the fourth rows of Table 1. For the third and the fourth systems, we submit the average model of last 20 epochs using the stochastic weight averaging. Note that all four systems are single model without any kind of ensemble methods applied [6].

7. REFERENCES

[1] 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

[2] J.-w. Jung, H.-S. Heo, H.-j. Shim, and H.-J. Yu, "Knowledge Distillation in Acoustic Scene Classification," *IEEE Access*, vol. 8, pp. 166 870–166 879, 2020.

[3] H.-S. Heo, J.-w. Jung, H.-j. Shim, and H.-J. Yu, "Acoustic scene classification using teacher-student learning with soft-labels," in *Annual Conference of the ISCA, INTERSPEECH*, 2019, pp. 614–618.

[4] J.-w. Jung, H.-S. Heo, H.-j. Shim, and H.-J. Yu, "Distilling the Knowledge of Specialist Deep Neural Networks in Acoustic Scene Classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2019, pp. 114–118.

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "MixUp: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[6] P. Izmailov, A. Wilson, D. Podoprikhin, D. Vetrov, and T. Garipov, "Averaging weights leads to wider optima and better generalization," in *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2018, pp. 876–885.

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

[8] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, *et al.*, "A two-stage approach to device-robust acoustic scene classification," *arXiv preprint arXiv:2011.01447*, 2020.

[9] H.-j. Shim, J.-h. Kim, J.-w. Jung, and H.-J. Yu, "Attentive max feature map for acoustic scene classification with joint learning considering the abstraction of classes," *arXiv preprint arXiv:2104.07213*, 2021.

[10] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[12] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[14] M. Kosmider, "Spectrum correction: Acoustic scene classification with mismatched recording devices," in *Annual Conference of the ISCA, INTERSPEECH*, 2020, pp. 4641–4645.

[15] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[16] 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," 2021.