MOBILENET USING COORDINATE ATTENTION AND FUSIONS FOR LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE DEVICES

Technical Report

Soonshin Seo, Ji-Hwan Kim†

Sogang University
Dept. of Computer Science and Engineering
Seoul, Repulic of Korea
{ssseo, kimjihwan}@sogang.ac.kr

ABSTRACT

In this technical report, we describe our acoustic scene classification methods submitted to detection and classification of acoustic scenes and events challenge 2021 task 1a. We extracted the log-Mel filter bank features with delta and delta-delta from the acoustic signals and applied normalization. A total of 6 data augmentations were applied as follows: mixup, spectrum augmentation, spectrum correction, pitch shift, speed change, and mix audios. In addition, we designed MobileNet using coordinate attention and fusions. Inspired by MobileNetV2, inverted residuals and linear bottlenecks are adapted for mobile blocks of the proposed MobileNet. We applied coordinate attention and early/late fusion methods after mobile blocks. In addition, we reduced the model size by applying weight quantization to the trained model. Experiments were conducted on the cross-validation setup of the official development set. We confirmed that our model achieved a log-loss of 1.040 and an accuracy of 72.6% within the 128 KB model size.

***Index Terms*—** Low-complexity acoustic scene classification, multiple devices, data augmentation, MobileNet, coordinate attention

1. INTRODUCTION

Acoustic scene classification (ASC) is a problem that takes an acoustic signal as input and classifies it into an appropriate acoustic scene. In particular, various research has been published for several years through the detection and classification of acoustic scenes and events (DCASE) challenge [1-3]. Specifically, the DCASE 2021 Challenge task 1a aims to classify a 10-second acoustic signal recorded by multiple devices [3]. At the same time, the model complexity limit of 128 KB is set for the non-zero parameters. As an evaluation metric, the average of the class-wise log loss is used along with the average of the class-wise accuracies.

In this technical report, we propose the following three methods. First, normalization and augmentation are applied to the log-Mel filter bank feature. Second, we propose MobileNet using coordinate attention and fusions. Finally, weight quantization is applied to the trained model for low-complexity. These are explained in Chapters 2 and 3, respectively. Section 4 shows the results for submission, and Section 5 concludes.

2. DATA PREPROCESSING AND AUGMENTATIONS

2.1. Datasets

The DCASE 2021 task 1a dataset consists of a development set and an evaluation set [2]. The acoustic scene classes in the dataset as follows: airport, shopping mall, metro station, street pedestrian, public square, street traffic, tram, bus, metro, and park.

As shown in Table 1, the development set consists of 10-second segments recorded with 3 real devices (A~C) and 6 simulated devices (S1~S6). The total duration and the number of segments are 64 hours and 23,040, respectively. As the cross-validation setup, the development set is split into 70% training set and 30% test set. In this case, several segments are not used for the balanced test set. Also, 3 simulated devices (S3~S6) are included only in the test set. The number of segments in the training/test sets is 13,965 and 2,970, respectively.

The evaluation set consists of 10-second segments recorded with 11 devices including 1 real device (D) and 4 simulated devices (S7~S11). The total number of segments is 7,920. The evaluation set is only used for submission.

Table 1: The overview of datasets.

Description	Devices	Segments
Dev. set (full)	9	23,040
Dev. set (cross-val., training)	6	13,965
Dev. set (cross-val., test)	9	2,970
Eval. set	11	7,920

2.2. Data Preprocessing

All audio segments are formatted with a mono channel, 44 kHz sampling rate, and 24-bit resolution per sample. For each 10-second input segment, 2048 FFT points were performed to every 1024 samples, and a power spectrum was extracted. That is, the number of bins of one power spectrum is 431.

† Corresponding author

Next, log-Mel filter bank features with 128 frequency bins were extracted, and mean and variance normalization was applied to each frequency bin. Also, delta and delta-delta were calculated from the normalized log-Mel filter bank features and stacked into the channel axis. Therefore, one input feature has the shape of 128×423×3.

2.3. Data Augmentations

Inspired by [4-6], the following data augmentation method was applied to the features: mixup [4], spectrum augmentation [5], spectrum correction [6], pitch shift, speed change, and mix audios.

Mixup and spectrum augmentation were used in the training process. For each mini-batch, the input features were randomly masked on the time and frequency axes, and then a mixup was applied with an alpha value of 0.4.

The other augmentation methods such as spectrum correction, pitch shift, speed change, and mix audios were applied before training. For spectrum correction, reference device spectrums were generated by averaging the spectrum from all training devices except device A. The spectrums of device A were corrected by using the reference device spectrum. In addition, the acoustic signals of all training datasets were augmented by randomly shifting the pitch and randomly changing the speed with padding and cropping. Also, the randomly mixing acoustic signals between the same classes were applied. As a result of data augmentations, the amount of the total training data set is increased as shown in Table 2.

Table 2: The comparison of data amount according to augmentation methods.

Description	Devices	Segments
Dev. set (full)	9	106,560
Dev. set (cross-val., training)	6	66,075

3. PROPOSED MOBILENET

3.1. Architectures

Inspired by MobNet [6] and MobileNetV2 [7], we designed two MobileNet. The one is MobileNet using coordinate attention [8] in Table 3, and the other is MobileNet using coordinate attention and fusions in Table 8. Hyperparameters of the proposed networks were determined by using grid search in the various experiments.

As shown in Table 3, the first proposed MobileNet mainly consists of mobile blocks and coordinate attention. The first 2-dimensional convolution layer and three mobile blocks are used to input features. Each mobile block consists of 32, 48, and 64 channels, and is designed to have wide channel dimensions. Then, batch normalization (BN) and ReLU activation functions are applied to the features. Next, after one convolution layer and dropout regularization with a 0.3 ratios, the coordinate attention is applied to the features. Finally, the features generated from coordinate attention are fed into the last convolution layer, and global average pooling (GAP) and softmax are applied.

Table 3: The architecture of the proposed MobileNet using coordinate attention.

Description	Configuration	Output shape
Input	-	128×423×3
Conv2D	32, 3×3, stride={2,2}	64×212×32
BN + ReLU	-	-
Mobile block	32, 3×3, stride={2,2}	32×106×32
Mobile block	48, 3×3, stride={2,2}	16×53×48
Mobile block	64, 3×3, stride={2,2}	8×27×64
Conv2D	64, 1×1, stride={1,1}	8×27×64
BN + ReLU	twice	-
Conv2D	64, 1×1, stride={1,1}	8×27×64
Dropout	0.3	-
Coordinate att.	$r = 4$	8×27×64
BN	-	-
Conv2D	10, 1×1, stride={1,1}	8×27×10
BN	-	-
GAP	-	1×10
Softmax	-	1×10

The inverted residuals and linear bottlenecks, proposed by MobileNetV2 [7], are applied to the mobile blocks of proposed MobileNet. As shown in Table 4, the mobile block consists of one bottleneck with stride 2 and two bottlenecks with stride 1. All bottlenecks are narrow-wide-narrow structures. The output features generated in the previous bottleneck are passed to the next bottleneck linearly without activations.

Table 4: The architecture of the mobile block.

Description	Configuration	Output shape
Input	-	$H \times W \times C_{in}$
Bottleneck	C_{out}, 3×3, stride={2,2}	$H/2 \times W/2 \times C_{out}$
Bottleneck-res.	C_{out}, 3×3, stride={1,1}	$H/2 \times W/2 \times C_{out}$
Bottleneck-res.	C_{out}, 3×3, stride={1,1}	$H/2 \times W/2 \times C_{out}$

Table 5: The architecture of the bottleneck.

Description	Configuration	Output shape
Input	-	$H \times W \times C_{in}$
Conv2D	$2C_{in}$, 1×1, stride={1,1}	$H \times W \times 2C_{in}$
BN + ReLU	-	-
Depthwise2D	$2C_{in}$, 3×3, stride={2,2}	$H/2 \times W/2 \times 2C_{in}$
BN + ReLU	-	-
Conv2D	C_{out}, 1×1, stride={1,1}	$H/2 \times W/2 \times C_{out}$
BN	-	-

Table 6: The architecture of the bottleneck-residual.

Description	Configuration	Output shape
Input	-	$H \times W \times C_{in}$
Conv2D	$2C_{in}$, 1×1, stride={1,1}	$H \times W \times 2C_{in}$
BN + ReLU	-	-
Depthwise2D	$2C_{in}$, 3×3, stride={1,1}	$H \times W \times 2C_{in}$
BN + ReLU	-	-
Conv2D	C_{out}, 1×1, stride={1,1}	$H \times W \times C_{out}$
BN	-	residual
Add	residual + input	$H \times W \times C_{out}$

As shown in Table 5 and Table 6, in the bottleneck with stride 2, the feature dimension is reduced by half through the depth-wise convolution layer. On the other hand, in the bottleneck with stride 1, it is trained while maintaining the feature dimension, and skip connections are applied. Also, all bottlenecks are applied to channel expansion at the first convolution layer and recovered at the last convolution layer.

3.2. Coordinate Attention

We adopted a novel attention mechanism for mobile networks by embedding positional information into channel attention named coordinate attention. Unlike squeeze-and-excitation channel attention [9], coordinate attention decomposes channel attention into two feature encoding using bi-directional average pooling. It can train the log-range dependencies and accurate location information in the feature maps [8].

As shown in Table 7, two 2-dimensional average pooling are used for the X and Y axes. Next, after the output features are concatenated, the number of channels is adjusted according to the value of the reduction ratio r. As the activation function after BN, swish activation using ReLU6 is used [8]. Then, it is split into X and Y axes to generate each attention weight. These attention weights are applied multiplication to the input features.

Table 7: The overview of coordinate attention.

Description	Configuration	Output shape
Input	-	H×W×C
AvgPool2D	1×W, stride={1,1}, H×1, stride={1,1}	1×W×C, H×1×C
Concat	-	(H+W)×1×C
Conv2D	C/r, 1×1, stride={1,1}	(H+W)×1×C/r
BN + Act.	-	-
Split	-	1×W×C/r, H×1×C/r
Conv2D	C, 1×1, stride={1,1}	1×W×C, H×1×C
Sigmoid	-	att. weights
Mul	input * att. weights	H×W×C

3.3. Fusions

The fusion methods, as well as coordinate attention, were applied for the proposed MobileNet. As shown in Table 8, two output features generated by different strides in the first convolution layer are fused (early fusion). Also, the output features of the last convolution layer are split in half. For the separated features, coordinate attention is applied to one side and is not applied to the other side. Then, GAP and softmax are applied to both output features, and the probability values are fusion (late fusion).

We confirmed through an experiment that these early and late fusions produced similar effects to the ensemble when applied together. We also applied various strides for the first convolutional layer. Unlike stride {2, 1}, which can be fuse along the time axis, stride {1, 2} can be fused along the frequency axis, and stride {2, 2} can be fused in both directions. In the case of split operation, it was confirmed that proper performance was obtained only when the axis of early fusion was the same.

Table 8: The architecture of the proposed MobileNet using coordinate attention and fusions with stride {2, 1}.

Description	Configuration	Output shape
Input	-	128×423×3
Conv2D	32, 3×3, stride={2,2}, 32, 3×3, stride={2,1},	64×212×32, 64×423×32
Early fusion	-	64×635×32
BN + ReLU	-	-
Mobile block	32, 3×3, stride={2,2}	32×318×32
Mobile block	48, 3×3, stride={2,2}	16×159×48
Mobile block	64, 3×3, stride={2,2}	8×80×64
Conv2D	64, 1×1, stride={1,1}	8×80×64
BN + ReLU	twice	-
Conv2D	72, 1×1, stride={1,1}	8×80×72
Dropout	0.3	-
BN	-	-
Conv2D	10, 1×1, stride={1,1}	8×80×10
BN	-	-
Split	-	8×40×10, 8×40×10
Coordinate Att. + GAP + Softmax	r = 8	1×10, out_A
GAP + Softmax	-	1×10, out_B
Late fusion	0.5*out_A + 0.5*out_B	1×10

3.4. Weight Quantization

Task 1a limits a model complexity to 128KB (only for non-zero parameters). We applied weight quantization to the trained model using Tensorflow-Lite converter. It can be converted from A 32-bit Tensorflow format to an 8-bit Tensorflow-Lite format.

3.5. Training Setup

All experiments in this paper were conducted using Tensorflow2.0 and Keras. The optimizer used the stochastic gradient descent with a 0.9 momentum weight and a 10^{-6} decay. Also, categorical cross-entropy loss was used. All our models were trained for 256 epochs with a batch size of 32. The initial learning rate was set to 0.1. At epochs 3, 7, 15, 31, 127, and 255, the learning rate was reset to obtain the re-training effect. We used the checkpoint with the lowest validation log-loss (or highest validation accuracy) as the best model. Our code is available at `https://github.com/sunshines14/DCASE2021`

4. RESULTS AND SUBMISSIONS

The experimental results and details of submissions can be confirmed in Table 9~11. We selected the following four models for submission among various models: proposed MobileNet using coordinate attention (tag 1), proposed MobileNet using coordinate attention and fusion with stride {2, 1} (tag 2), proposed MobileNet using coordinate attention and fusion with stride {2, 2} (tag 3), proposed MobileNet using coordinate attention and fusion with stride {1, 2} (tag 4).

Table 9: The overall performances of submissions.

Description	Size	Loss	Acc.	Tag
Official baseline	90.3	1.473	47.7	-
Coordinate att.	125	**1.040**	69.0	1
Fusion stride 21	126.5	1.089	**72.6**	2
Fusion stride 22	126.6	1.092	72.1	3
Fusion stride 12	126.5	1.106	**72.6**	4

Table 10: The class-wise log-losses of submissions.

Class / Tag	1	2	3	4
Airport	1.504	**1.138**	1.248	1.360
Bus	**0.708**	0.790	0.744	0.869
Metro	**0.892**	1.063	1.022	1.135
Metro station	**0.914**	1.134	1.115	1.101
Park	**0.703**	0.837	0.893	0.827
Public square	1.434	**1.304**	1.420	1.400
Shopping mall	1.231	1.098	**1.032**	1.161
Street pedestrian	1.475	1.550	1.659	**1.457**
Street traffic	**0.502**	0.785	0.653	0.681
tram	**0.948**	1.193	1.130	1.064

Table 11: The class-wise log-losses of submissions.

Device / Tag	1	2	3	4
A	**0.985**	0.978	0.984	1.018
B	**1.079**	1.156	1.111	1.164
C	**1.010**	1.034	1.080	1.064
S1	**1.026**	1.110	1.128	1.102
S2	**1.050**	1.125	1.124	1.109
S3	**1.045**	1.108	1.080	1.108
S4	**1.035**	1.070	1.093	1.106
S5	**1.070**	1.090	1.106	1.137
S6	**1.061**	1.134	1.117	1.143

5. CONCLUSION

This technical report aims to describe our low-complexity ASC models for DCASE 2021 task 1a. We extracted log-Mel filter bank features and applied normalization/augmentations. We designed MobileNet using coordinate attention and fusions and applied weight quantization. Experiments were conducted on the cross-validation setup of the official development set. We confirmed that our model achieved a log-loss of 1.040 and an accuracy of 72.6% within the 128 KB model size.

6. ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIT/IITP [2017-0-00050, Development of Human Enhancement Technology for Auditory and Muscle Support].

7. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proc. DCASE Workshop*, pp. 2018, 9-13.

[2] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions,” in *Proc. DCASE Workshop*, 2020.

[3] I. Martin-Morato, 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.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.

[5] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: a simple data augmentation method for automatic speech recognition,” in *Proc. ISCA Interspeech*, 2019, pp. 2019-2680.

[6] H. Hu, C-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C-H. Lee, “Device-robust acoustic scene classification based on two-stage categorization and data augmentation,” in *Proc. DCASE Challenge*, 2020.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L-C. Chen, “MobileNetV2: inverted residuals and linear bottlenecks,” in *Proc. IEEE CVPR*, 2018, pp. 4510-4520.

[8] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” *in Proc. IEEE CVPR*, 2021.

[9] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE CVPR*, 2018, pp. 7132-7141.