LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION BASED ON RESIDUAL NET

Technical Report

Jiangnan Liang, Cheng Zeng, Chuang Shi, Le Zhang, Yisen Zhou, Yuehong Li, Yanyu Zhou, Tianqi Tan

School of Information and Communication Engineering,
University of Electronic Science and Technology of China, Chengdu, China
liangjiangnan@std.uestc.edu.cn, shichuang@uestc.edu.cn

ABSTRACT

This technical report describes the submitted systems for task 1 of the DCASE 2022 challenge. The log-mel energies, delta features and delta-delta features were extracted to train the model. We adopted a total of eight data augmentation methods. BC-ResNet and MobileNetV2 were used as training model. We used knowledge distillation and quantization to compress the model. Our systems achieved lower log loss and higher accuracy in the development dataset than the baseline system.

Index Terms— DCASE 2022, acoustic scene classification, data augmentation, BC-ResNet, MobileNetV2, model compression

1. INTRODUCTION

In DCASE2022, task 1 focused on classifying acoustic scenes with mismatched recording devices. It implified that some devices with only appear in the evaluation dataset. As compared to DCASE2021, the model size is further restricted. The goal is to build a ten-class classifier whose maximum number of parameters is 128K and maximum number of MACS per inference is 30 MMAC. Moreover, the duration of each audio in the DCASE2022 dataset has been shortened to 1 second. This reduction of audio length certainly increases the challenges for this task.

Based on these two changes, we proposed our low-complexity acoustic scene classification systems. We chose to extract log-mel energies[1] and generate delta features and delta-delta features. To augment the acoustic features, we adopted eight methods of data augmentation.

2. ARCHITECTURE

2.1. Network Architecture

2.1.1. Broadcasting-residual network

In DCASE2021, broadcasting-residual network(BC-ResNet) proposed by Kim et al.[2] achieved good results. Our proposed model modified based on BC-ResNet, is shown in Table 1. The architecture of broadcasting-residual normal block (BC-ResNormBlock) and transition block (BC-ResTransBlock) are shown in Fig. 1[3].

We also used Residual Normalization(ResNorm) in input features and input of Transition blocks architecture for better domain generalization[2, 4, 5]. The number of channels c is set to 80. The number of parameters is 314.8K (maximum is 128K in system complexity requirement) and the number of MACS per inference is 998.527 MMAC (maximum is 30 MMAC in system complexity requirement).

Table 1: The architecture of the proposed BC-ResNet.

Operator	Input Shape	Out Channels
Conv2d 5×5, stride 1	256×44×1	2c
BC-ResTransBlock	256×44×2c	c
BC-ResNormBlock	256×44×c	c
BC-ResTransBlock	256×44×c	1.5c
BC-ResNormBlock	256×44×1.5c	1.5c
BC-ResTransBlock	256×44×1.5c	2c
BC-ResNormBlock	256×44×2c	2c
BC-ResTransBlock	256×44×2c	2.5c
BC-ResNormBlock	256×44×2.5c	2.5c
BC-ResNormBlock	256×44×2.5c	2.5c
Conv2d 1×1	256×44×2.5c	10
Avgpool	256×44×10	10
Softmax	1×10	10

2.1.2. MobileNetV2

The second network we used is MobileNetV2 combined with coordinate attention[6, 7, 8]. The model structure and the number of parameters is provided in Table 2. The last column is the number of parameters corresponding to network layer. The number of parameters can be obtained by adding up the last column, which numbers 110,452 parameters.

The specific structure of bottleneck and bottleneck-res which appeared in Table 2 is provided in Table 3 and Table 4 respectively. More details regarding coordinate attention are shown in Fig. 2[8].

2.2. Model Compression

To compress the proposed model, two model compression schemes are used: knowledge distillation and quantization.

Knowledge Distillation: The BC-ResNet (c=80) is set to teacher model. The student model uses a similar architecture with some modifications, and the c is set to 40. KD loss is used to make the performance of the student model to be closer to that of teacher model[9].

Quantization: We used 8-bit quantization method after training the model.

After compressing the BC-ResNet, the number of parameters is shown in Table 5. The number of parameters can be obtained by adding up the last column, which is 85,800 parameters. The number of parameters of first BC-ResTransBlock and BC-ResNormBlock are concretely shown in Table 6 and Table 7. The remaining blocks only provided the total number of parameters and will not be de-

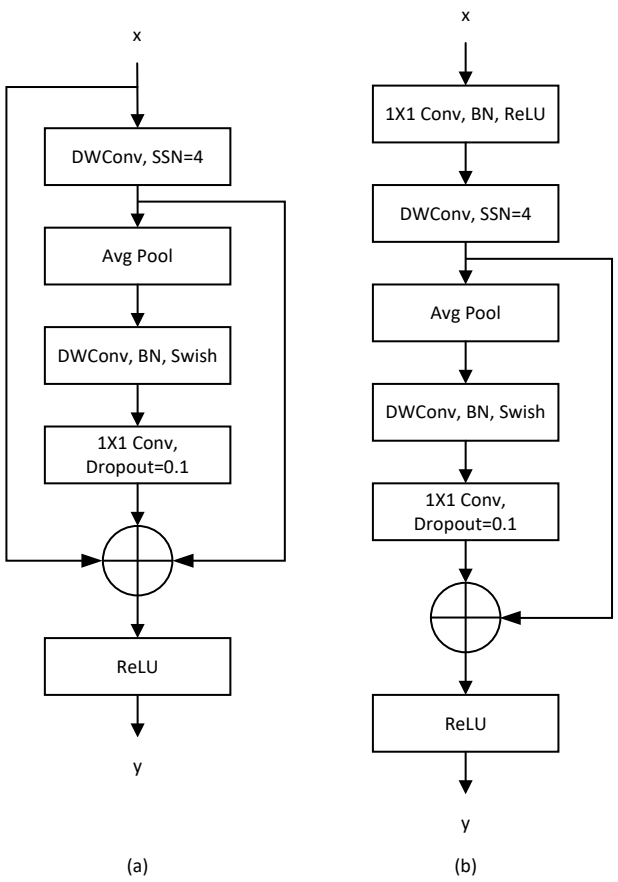

Figure 1: (a) The architecture of BC-ResNormBlock; (b) The architecture of BC-ResTransBlock.

scribed concretely. The MobileNetV2 is also quantified as the task requirement.

2.3. Acoustic Features Extraction

In BC-ResNet, the log-mel energies is used as acoustic features. We used Short Time Fourier Transform with a hamming window, whose size is 2048 and overlap is 50%. Afterwhich the log mel filter bank with 256 log mel filters is used to obtain the log-mel energies. Therefore, the size of the input feature of BC-ResNet is (256,44,1). For MobileNetV2, there are 128 log mel filters in the filter bank. Moreover, the delta features and delta-delta features are added to form three-channel features, whose shape is (128,36,3).

2.4. Data Augmentation

To avoid over-fitting and to improve the generalization ability of the model, we applied eight methods of data augmentation, including noise addition[10], pitch shifting[11], speed changing[12], time masking[13], frequency masking[13], time warping[13], frequency warping, and mix-up[14]. MobileNetV2 uses five methods, including noise addition, pitch shifting, speed changing, time masking, and mix-up. All systems of BC-ResNet include time masking, frequency masking, time warping, and mix-up. Only one system of BC-ResNet includes frequency warping.

Table 2: The architecture of MobileNetV2 combined with coordinate attention.

Description	Configuration	Outshape	Param
Input	-	128×36×3	0
Conv2D	32,3×3,stride=[2,2]	64×18×32	896
BN + RELU	-	64×18×32	128
Bottleneck	32,3×3,stride=[2,2]	32×9×32	5,472
Bottleneck-res	32,3×3,stride=[1,1]	32×9×32	5,472
Bottleneck-res	32,3×3,stride=[1,1]	32×9×32	5,472
Bottleneck	48,3×3,stride=[2,2]	16×5×48	6,576
Bottleneck-res	48,3×3,stride=[1,1]	16×5×48	11,280
Bottleneck-res	48,3×3,stride=[1,1]	16×5×48	11,280
Bottleneck	64,3×3,stride=[2,2]	8×3×64	12,896
Bottleneck-res	64,3×3,stride=[1,1]	8×3×64	19,136
Bottleneck-res	64,3×3,stride=[1,1]	8×3×64	19,136
Conv2D	64,1×1,stride=[1,1]	8×3×64	4,160
BN + RELU	-	-	256
BN + RELU	-	-	128
Conv2D	64,1×1,stride=[1,1]	8×3×64	4,096
Dropout	0.3	-	0
Coordinate attention	r=4	8×3×64	3,280
BN	-	-	128
Conv2D	10,1×1,stride=[1,1]	8×3×10	640
BN	-	-	20
GAP	-	1×10	0
Softmax	-	1×10	0

Table 3: The architecture of bottleneck.

Description	Configuration	Outshape
Input	-	H×W×C_{in}
Conv2D	2C_{in},1×1,stride=[1,1]	H×W×2C_{in}
BN +RELU	-	-
Depthwise2D	2C_{in},3×3,stride=[2,2]	H/2×W/2×2C_{in}
BN +RELU	-	-
Conv2D	C_{out},1×1,stride=[1,1]	H/2×W/2×C_{out}
BN	-	-

3. EXPERIMENTS

3.1. Datasets

The DCASE 2022 task 1 dataset is divided into two parts. The development dataset contains 1-second segments recordings from 12 European cities in 10 different acoustic scenes using 4 different devices (3 real devices A-C and 6 simulated devices S1-S6). Additionally, synthetic data for 11 mobile devices is created based on the original recordings. The acoustic scene classes in the dataset are as follows: airport, shopping mall, metro station, street pedestrian, public square, street traffic, tram, bus, metro, and park. The development dataset comprises 40 hours of data from device A, and smaller amounts from the other devices. Audio is provided in a single-channel 44.1kHz 24-bit format. The dataset is provided with a training/test split in which 70% of the data for each device is included for training, 30% for testing. 3 simulated devices (S3-S6) are included only in the test set.

The evaluation set consists of 1sec segments recorded with 11 devices including 1 real device (D) and 4 simulated devices (S7-S11). The evaluation set is only used for submission.

Table 4: The architecture of bottleneck-residual.

Description	Configuration	Outshape
Input	-	H×W×C_{in}
Conv2D	$2C_{in}$,1×1,stride=[1,1]	H×W×$2C_{in}$
BN +RELU	-	-
Depthwise2D	$2C_{in}$,3×3,stride=[1,1]	H×W×$2C_{in}$
BN +RELU	-	-
Conv2D	C_{out},1×1,stride=[1,1]	H×W×C_{out}
BN	-	residual
Add	residual+input	H×W×C_{out}

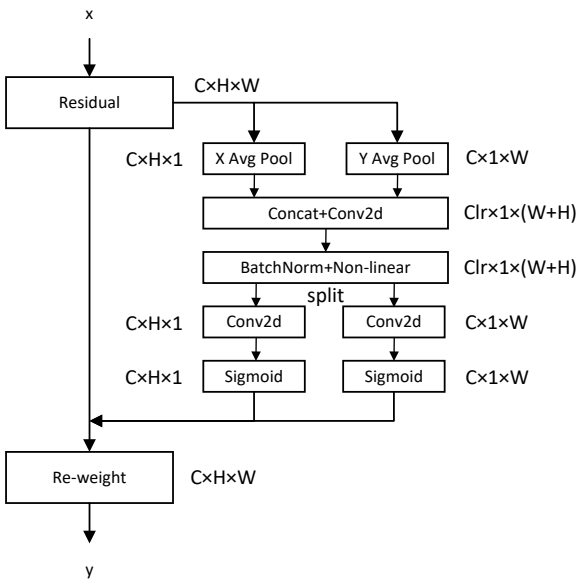

Figure 2: The architecture of coordinate attention.

3.2. Implementation Details

In BC-ResNet, the models are trained for 120 epochs with a batch size of 32, using stochastic gradient descent optimizer with a 0.9 momentum and a 10^{-3} weight decay. The learning rate starts with 0.008 and multiplies by a reduction ratio of 0.4 if loss does not fall in 3 epochs. For knowledge distillation, the learning rate increases from 0 to 0.006 as a warmup in the first ten epochs[15]. Then, it reduces to zero with cosine annealing [16]. MobileNetV2 is trained for 64 epochs with a batch size of 32, using SGD optimizer with a 0.9 momentum and a 10^{-6} weight decay. The learning rate starts with 0.1 and reduces to zero with cosine annealing [16]. While at epochs 3, 7, 15, 33, the learning rate is reset to 0.1 again and the damping period is reset.

3.3. Results

Table 8 lists the models we submitted. It also includes the multiclass cross-entropy (Log loss) in the development dataset, the average accuracy of scene classification, the number of parameters, and the number of MACS per inference. The baseline system is compared in the table. The BC-ResNet1 did not use ResNorm. Moreover, four data augmentation methods were used, including time masking, frequency masking, time warping, and mix-up. The BC-ResNet2 both used ResNorm and the four methods mentioned above. While the BC-ResNet3 both used ResNorm and five data augmentation methods. Four of them were the same as above, with frequency warping used additionally.

Table 5: The number of parameters of the student model.

Operator	Input Shape	Out Channels	Param
Conv2d 5×5, stride 2	256×44×1	2c	2080
BC-ResTransBlock	128×22×2c	c	5520
BC-ResNormBlock	128×22×c	c	2240
MaxPool 2×2	128×22×c	c	-
BC-ResTransBlock	64×11×c	1.5c	7080
BC-ResNormBlock	64×11×1.5c	1.5c	4560
MaxPool 2×2	64×11×1.5c	1.5c	-
BC-ResTransBlock	32×5×1.5c	2c	12640
BC-ResNormBlock	32×5×2c	2c	7680
MaxPool 2×2	32×5×2c	2c	-
BC-ResTransBlock	16×2×2c	2.5c	19800
BC-ResNormBlock	16×2×2.5c	2.5c	11600
BC-ResNormBlock	16×2×2.5c	2.5c	11600
Conv2d 1×1	16×2×2.5c	10	1000
Avgpool	16×2×10	10	-
Softmax	1×10	10	-

Table 6: The number of parameters of the first BC-ResTransBlock.

Operator	Param
Conv2d	3200
BatchNorm2d	80
ReLU	-
Conv2d	120
SubSpectralNorm	-
BatchNorm2d	320
Conv2d	120
BatchNorm2d	80
SiLU	-
Conv2d	1600
Dropout2d	-
ReLU	-

4. CONCLUSION

In this report, we described our systems for DCASE2022 task 1. We used log-mel energies, delta features and delta-delta features as acoustic features and adopted eight methods of data augmentation. We used knowledge distillation and quantization to compress the models to achieve task requirement. Our models were within the 128K parameters and 30 MMAC, and achieved lower log loss.

5. REFERENCES

[1] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (mlsa) filter for speech synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[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.

Table 7: The number of parameters of the first BC-ResNormBlock.

Operator	Param
Conv2d	120
SubSpectralNorm	-
BatchNorm2d	320
Conv2d	120
BatchNorm2d	80
SiLU	-
Conv2d	1600
Dropout2d	-
ReLU	-

Table 8: Results for the development dataset.

Model	Log loss	Accuracy	Param	MACS
Baseline	1.575	42.9%	46,512	29.235M
BC-ResNet1	1.263	53.8%	85,800	20.5M
BC-ResNet2	1.267	55.9%	85,800	20.5M
BC-ResNet3	1.236	56.2%	85,800	20.5M
MobileNetV2	1.5565	45.86%	110,452	11.186M

[3] J. L. Byeonggeun Kim, Simyung Chang and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proceedings of the Interspeech*, 2021, pp. 4538–4542.

[4] H. Nam and H. E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[5] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[6] S. Seo and J. H. Kim, "Mobilenet using coordinate attention and fusions for low-complexity acoustic scene classification with multiple devices," DCASE2021 Challenge, Tech. Rep., June 2021.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[8] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 713–13 722.

[9] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," in *2020 25th International Conference on Pattern Recognition*, 2021, pp. 4619–4625.

[10] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, vol. 346, p. 108885, 2020.

[11] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proceedings of the International Society for Music Information Retrieval*, 2015, pp. 121–126.

[12] H. Hu, C. H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, *et al.*, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," *arXiv preprint arXiv:2007.08389*, 2020.

[13] 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," *arXiv preprint arXiv:1904.08779*, 2019.

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

[15] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[16] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.