KT SUBMISSION FOR THE DCASE 2022 CHALLENGE: MODERNIZED CONVOLUTIONAL NEURAL NETWORKS FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

TaeSoo Kim, Gahui Lee, JaeHan Park

KT Corporation, South Korea

ABSTRACT

In this technical reports, we present our team's submission for DCASE 2022 TASK1 which is the low complexity Acoustic Scene Classification (ASC). We gradually modernized a neural network architecture design starting from the baseline model and discover several key components that contribute to the performance. To meet constraints of the model complexity, the number of parameters and the number of MACs are considered while applying each designs. As a result, our model achieves 1.2593 log-loss and 54.03% accuracy on the development set, while having less than 114k of total parameters (including the zero-valued) and 30 million MACs.

Index Terms— acoustic scene classification, ConvNeXt, CNN

1. INTRODUCTION

Detection and Classification of Acoustic Scenes and Events (DCASE)[1] is an annual challenge for people interested in environmental sound classification and detection. We focus on TASK 1, the low complexity acoustic scene classification(ASC), which is the task of classifying a recording into one of the predefined ten acoustic scene classes such as *airport*, *shopping mall*, *metro station*, *pedestrian street*, *public square*, *street traffic*, *tram*, *bus*, *metro* and *park*. This is the same as the last year challenge, but it is different in that the length of data is 10sec to 1sec and data type of model weights is fixed into INT8 and maximum number of parameters (including zero-valued) and MACs is limited to respectively 128K and 30 million.

To find the neural network architecture to meet these constraints and demonstrate the best performance, we adopted a method of deriving the optimal structure by gradually changing the variable of model structure and all hyper-parameters affecting the performance inspired by [2]. Starting from the baseline system provided by the challenge, we first investigated acoustic features suitable for audio task and explored the various training procedures as well as augmentation methods. Based on these fixed investigated results, we changed the model architecture using ResNet Block[3] which is achieved successful result in the image field and inspected the combination of kernel size and stride used in stem cell layer related to the number of MACs. And then, we progressively modify the model architecture by using blocks such as ResNeXt[4] block and inverted bottleneck block[5] to be more modernized. We entirely probed the kernel size except the stem layer, several activation functions, and normalization methods to find out which method has a good impact on the performance. After that, we adopt the self attention based pooling layer widely used in speaker recognition field and seperate downsample layer for adjusting feature map sizes.

Designs	levels	num params	MMACs	Acc(%)	log-loss
Baseline	-	46,118	29.272	44	**1.52**
ResNet	-	60,566	26.584	44	**1.51**
Stem	-	78,582	29.821	47	**1.46**
ResNext	-	50,574	29.389	48	**1.43**
Inverted	-	33,735	28.854	48.9	**1.3882**
Kernel Size	3	30,210	26.391	48.46	1.3994
	5	30,953	27.151	48.92	**1.3816**
	7	33,280	29.466	47.87	1.4239
Activation	GELU	30,953	27.151	48.92	1.3516
	Swish			50.57	**1.3451**
	PReLU			47.87	1.4172
Few activations	-	30,953	27.151	52.18	**1.3362**
SAP	-	113,318	29.481	53.91	**1.2841**
s.d conv	-	113,438	29,481	54.03	**1.2593**

Table 1: Performance improvements for each network model design applications. Bold texted on log-loss refers to the design adopted at each stage

The rest of the report is organized as follows. In section 2, the proposed method and the detailed experimental environment are explained. Submitted systems and conclusion are included in section 3 and 4 respectively.

2. METHOD

In this section, we provide a roadmap from the baseline model to the proposed method. As [2], We gradually applied a series of model architecture designs with the baseline model as the starting point. The applied design is adopted if there is an improvement in log loss. The number of primary channels is adjusted to meet the task limit of model complexity according to the number of MACs and parameters.

2.1. Acoustic Features and Training Procedures

Input acoustic features and training procedures affect the ultimate performance of neural networks. Therefore, both of input acoustic features and training procedures are fixed to measure only performance improvements by the network architecture designs. For the training techniques, we trained each designed network architecture for 30 epochs using AdamW optimizer [6] with mini-batch size to 32, momentum to 0.9, weight decay to 0.001, learning rate linearly increasing 0 to 0.003 for five warm-up epochs [7] before applying cosine annealing learning rate schedule [8]. We also used Specaugment [9] with two temporal masks and two frequency masks with 2 and 8 mask parameters, respectively. Dropout rate is set to 0.3. The official development set [1] and train/test setup are used for all experiments. The train split consists of 139,970 segments from all

real devices and three simulated devices (S1-S3) while the test split consists of 29,680 segments from all devices.

According to the our investigation, log Mel spectrograms are most commonly used acoustic features for the ASC task. Thus, the log Mel spectrograms are used for the input features. Plus, We did downsampling by 22kHz and used 80-dimensional log Mel spectrograms with a 40ms window length and a 20ms hop length. When training the baseline model architecture, we found that this settings performed best while satisfying the constraints. **In this setup, there was an improvement in log-loss from 1.53 to 1.52.**

2.2. ResNet Block

According to our survey, many participants proposed ResNet-based neural network designs [3] and ranked high in previous competitions of the ASC. We replaced last two convolution layers of the baseline model with two ResNet blocks. **With the modification, there was an improvement in log-loss from 1.52 to 1.51 so using ResNet block is selected.**

2.3. Stem Cell

The resolution size is closely correlated to the number of MACs. In general, the first convolution layer is related to actively downsampling the input with an appropriate feature map size. Therefore, we tried to find the best combination of the kernel size and stride of the stem cell. Experiments were conducted on the groups of three, four and five kernel sizes and on the candidate groups of one, two and three strides for each height and width. **As the result, the combination of five kernel sizes, one wide stride, and two high strides is chosen because it achieves a 1.46 log-loss which is better than the previous step.**

2.4. ResNeXt

ResNeXt [4] block have advantages of using grouped convolutions to balance between performance and model complexity. As [2], we also adopted depth-wise convolution layer [10] with same number of input channels and groups. **It achieves 1.43 log-loss by replacing ResNet blocks with ResNeXt blocks.**

2.5. Inverted Bottleneck

Here we also explore the inverted bottleneck [5]. Unlike conventional bottleneck block of gradually decreasing the number of filters, the inverted bottleneck increases the number of filters in the middle layer. In order to find an appropriate expansion ratio, the experiments were conducted by adjusting the expansion ratio from 1.5 to 4. **As the result, when the expansion ratio was set to 2, it showed the best performance at 1.39 log-loss.**

2.6. Kernel Sizes

We experiment with the kernel sizes including 3, 5 and 7. **As a result, kernel size 5 gets the best score at 1.38 log-loss than others.**

2.7. Activation Functions

Here we explore the activation functions including *ReLU* [11], *PReLU* [12] with the alpha to 0.2, *GELU* [13] and *Swish* [14] activation functions. **In comparison, *Swish* achieves 1.345 log-loss, showing the best score compared to other functions.** Plus, we also explore the number of activation functions used in the bottleneck blocks. As a result of experimenting with removing the activation functions of each convolution layer in the block one by one, **when the first and last activation functions were removed, there was an improvement by achieving 1.3362 log-loss.**

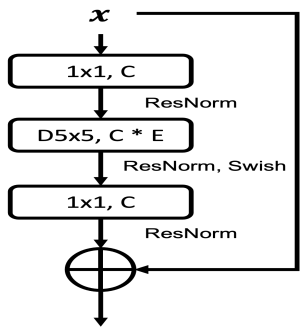

Figure 1: Final design of bottleneck blocks

2.8. Normalizations

Kim et al [15] proposed a novel normalization technique called *ResNorm* and showed significant improvements of performance on the ASC task. Inspired by this, we experiment with the normalization techniques including *batch normalization* [16], *layer normalization* [17], *ResNorm* [15] and *FreqIN* [18]. **As a result of the experiment, *ResNorm* achieved 1.32 log-loss, showing the best performance among them.** Experiments were conducted to reduce the number of normalization functions, but there was no performance improvement.

It shows how the block was designed up to this point. **Figure 1.** illustrates the block design we propose.

2.9. Pooling Layer

Since the size of the time axis is variable, it is common to fix the its length by pooling the feature map by the time axis and use it as input of linear layers to make decisions. The baseline model uses the *global average pooling* (GAP) [19] by the time axis. Instead of GAP, we conducted an experiment with using the *self attentive pooling* (SAP) [20]. Plus, a grouped convolution is adopted to SAP with same number of its input channels and groups. **Using the SAP, the log-loss improved from 1.32 to 1.2841.**

2.10. Seperate Downsample Layer

The seperate downsample layer is adopted to adjust the feature map sizes in [2], instead of pooling layers. Likewise, we conducted an experiments with adopting the seperate downsample layer. However, unlike [2], The *batch normalization* was performed instead of the *layer normalization* before convolution layer performed. **The seperate downsample layer leads an improvement of the log-loss to 1.259.**

Lastly, **Figure 2.** illustrate overall architecture of the proposed model, where k indicates kernel sizes, C indicates the number of channels, E indicates the expansion rate of the block.

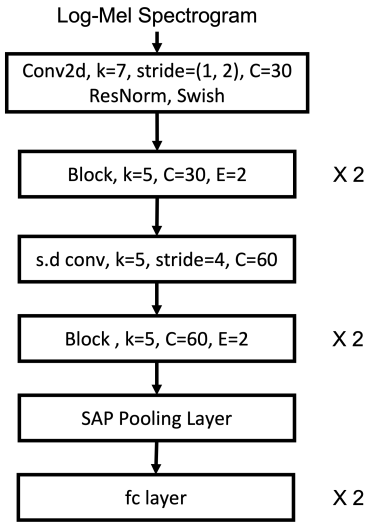

Figure 2: Overall architecture of the proposed network design

3. SUBMITTED SYSTEMS

For the submissions, we retrained the proposed neural network design with same procedures as experiments but the train/test data setup and the number of epochs. Since, a part of audio streams recorded by three simulated devices (S4-S6) are not included in the official data split fold, we added them into the database for training. To meet the task constraints that the weights of the network should be INT8 data type, the model is extra-trained with quantization-aware training (QAT) [21] with SGD optimizer for ten epochs.

4. CONCLUSION

In this work, we propose a novel neural network architecture for ASC gradually modernized as its performance improved while satisfying the constraints of model complexity, model containing less than 128k of total parameters (including zero-valued) and less than 30 million MACs. Quantization-Aware training is also adopted to meet the constraint that the weights of the submitted neural network should be fixed in INT8. Our model achieves 1.2593 log-loss and 54.03% accuracy improving 17% and 22.8% over the baseline model, respectively.

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

[2] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *arXiv preprint arXiv:2201.03545*, 2022.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[5] 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.

[6] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[7] 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.

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

[9] 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.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[12] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[13] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

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

[15] 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.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

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

[19] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[20] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *arXiv preprint arXiv:1602.03609*, 2016.

[21] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.