QTI SUBMISSION TO DCASE 2020: MODEL EFFICIENT ACOUSTIC SCENE CLASSIFICATION

Technical Report

Simyung Chang *Janghoon Cho* *Hyoungwoo Park* *Hyunsin Park*
Sungrack Yun *Kyuwoong Hwang*

Qualcomm AI Research†, Qualcomm Korea YH,
{simychan, janghoon, hwoopark, hyunsinp, sungrack, kyuwoong}@qti.qualcomm.com

ABSTRACT

This technical report describes the details of our submission for Task1B of the DCASE 2020 challenge. In this report, we introduce three methods for the efficient acoustic scene classification with low model complexity. First, inspired by CutMix which is proposed for image recognition tasks, we consider FreqMix for the data augmentation of mixing specific frequency bands of two different samples instead of cutting and pasting box patches. Second, as a novel feature normalization, we consider SubSpectral Normalization, which can adjust the information imbalance between each frequency band without increasing model size. Last, to reduce the number of model parameters, we propose a Shared Residual architecture where the weights of all layers (except the normalization layer) are shared. All submission models were trained without any external data, and our model is not based on an ensemble of multiple models but a single model to satisfy the model complexity condition.

Index Terms— Acoustic Scene Classification, FreqMix, SubSpectral Normalization

1. INTRODUCTION

The acoustic scene classification (ASC) has received a great attention in the field of acoustic signal processing. The goal of ASC is to predict the audio scene label of an environment sound [1], and it can be adopted to various applications such as context-awareness and surveillance [2, 3, 4] where the device recognizes the environmental sound by analyzing the input audio. Every year, we have the challenge of Detection and Classification of Acoustic Scenes and Events (DCASE) [5, 6] where the tasks of various ASC problems with large-scale datasets are introduced. And, there is an increasing number of research centers, companies, and universities who participate to this challenge and workshop.

In this year, a new task, "Low-complexity ASC", has been introduced and included in the challenge with the motivation that the device should classify acoustic scenes with only minimum resources and no communications between a cloud server and the device. In previous challenges, most algorithms including state-of-the art are based on a complex model whose size is usually more than 1 mega bytes [7, 8, 9]. Last year, we proposed an ASC architecture of Task1A [10] for the performance improvement with a smaller model due to the overfitting problem. The new task of this year, Task1B, restricts the size of classifier classifier to 500KB for the non-zero parameters [11]. This model size limit does not require much computation resources and memory, and thus the other jobs running on the device would not be affected by the ASC application in background.

For the Task 1B, the ASC model needs to be memory-efficient and also show good accuracy. To improve the performance of ASC, a number of researches have been published. For the ASC model architecture, a modified version of CNNs such as VGG or ResNet are widely used [12, 8, 10]. For the generalization of the model, data augmentation techniques such as MixUp or generative methods [13, 7] are considered. In [14, 10], receptive fields are used as a regularizer, and frequency-aware structred are applied. However, the previous studies do not consider the low model complexity. In this work, we consider two modules to achieve the high performance with small networks. The first module is *Shared Residual Block* which allows multiple layers to share their weights to increase the expressive power with fewer parameters. Here, we expect that an additional generalization can be obtained through the weight share. The second module is *SubSpectral Normalization* which reduces the correlation between frequency bands and also adjusts the information imbalance between each band without increasing model size.

In addition, we introduce a data augmentation technique, *FreqMix*, which is inspired by CutMix proposed for image recognition. In contrast to the CutMix where the box-shaped pathces are cut and pasted, the FreqMix creates new data by mixing the frequency band of a specific range from two different audio inputs. By combining the FreqMix with CutMix, our ASC model could improve the performance more. For the model compression, we adopt the L1 unstructured pruning [15] and half-precision floating-point. Also, we apply a class-balanced focal loss [16] which adjusts the data imbalance of the DCASE 2020 dataset. Finally, we achieved 98% validation accuracy with the model size less than 500KB.

The rest of the report is organized as follows. Section 2 describes the network architecture. Section 3 explains the data augmentation technique that we used. Section 4 explains our training loss. Section 5 describes network compression technique that we used. Section 6 shows the experimental results and analysis. Finally, Section 7 concludes our results.

2. NETWORK ARCHITECTURE

The goal of the model architecture design is to achieve the high performance with low model complexity. For this, we propose an architecture which applies several novel structures to the vanilla ResNet [17] which has been widely used in ASC task. Also, we ad-

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Table 1: Overall Network Architecture. c is the size of base channel. SSN(S) denotes SubSpectral Normalization with the number of subspectral group S, and SRB is the Shared Residual Block.

	Settings	Channels
Conv	5x5, stride=2	c
RB 1	3x3, BN, 1x1, SSN(S=2)	c
	2x2 max pooling	c
RB 2	3x3, BN, 3x3, SSN(S=2)	c
	2x2 max pooling	c
RB 3	3x3, BN, 3x3, SSN(S=2)	c
RB 4	3x3, BN, 3x3, SSN(S=2)	c
	2x2 max pooling	c
RB 5	1x1, BN, 1x1, SSN(S=2)	2c
SRB 1	N=3 Shared(1x1, BN, 1x1, BN)	2c
RB 6	1x1, BN, 1x1, SSN(S=2)	4c
SRB 2	N=3 Shared(1x1, BN, 1x1, BN)	4c
Conv	1x1, BN	#classes
Pool	Global Average Pooling	#classes

just the receptive field for the regularization as used in [18]. In the following subsections, we describe the details of our novel structures, and Table 1 shows the specification of the entire network architecture.

2.1. Shared Residual Architecture

The Residual Block in ResNet tries to learn the residual information which is added to the identity shortcut to model the desired output. These shortcut make ResNet behave like ensembles in relatively shallow networks. Removal of certain residual blocks usually has a slight impact on performance [19]. Based on this observation, we consider a *Shared Residual Block* where the weights of multiple residual blocks are shared to reduce the number of parameters while the expressive power is increased. Here, the shared residual blocks have the same input and output channels of the same spatial size. The batch normalization, however, is not shared and separately defined for each layer since the feature distribution of each layer is different due to the residual addition. Figure 1 shows our proposed network architecture consisting of N shared residual blocks. Compared to the ResNet with the same depth, the shared architecture reduces the model complexity by more than 30% without performance degradation.

2.2. SubSpectral Normalization

The proposed model consists of multiple 2D convolutions and takes Mel spectrogram as input. In most approaches based on the 2D convolution for audio data processing, the VGG or ResNet architecture have been widely used [12, 8, 10]. In image processing, the features can be obtained by applying 2D convolution to all spatial dimensions (e.g. height, width) of the input raw image. However, in the audio case, the Mel spectrogram input has different and unique characteristics in frequency domain, and thus the 2D convolution applying equally to the frequency and time dimension may not extract a good feature for audio scene classification. In this work, we introduce a *SubSpectral* normalization method which splits the input frequency dimension into several groups (sub-bands) and performs a different normalization for each group. Then, we apply the conventional 2D convolution to the normalized spectrum features. Figure 2 shows the comparisons of various normalization methods.

An affine transform can be applied to the normalization. Although we can consider a different affine transform for each frequency group, this work uses the same transform for all frequency groups.

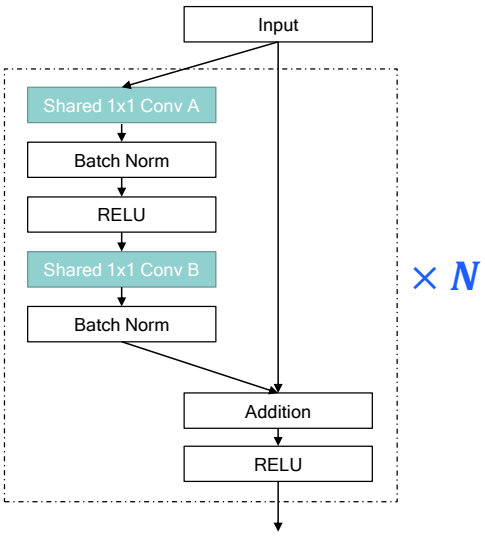

Figure 1: The Structure of Shared Residual Block. This block is used N times repeatedly.

3. DATA AUGMENTATION

To alleviate the overfitting in training our model, we mainly consider two data augmentation methods: Mixup[20] and FreqMix. The Mixup generates a new pair of training data and label by mixing two existing samples and corresponding labels with a certain ratio. Motivated by CutMix[21], we propose FreqMix which replaces a frequency band of an input spectrogram with that of other sample's, and the corresponding labels are also mixed with the same ratio of mixed frequency band

3.1. FreqMix

As described in sec. 2.2, the audio spectrum has different and unique characteristics in frequency bands, and thus we consider the FreqMix where a new pair of $(\hat{x}, \hat{y})$ is generated from two different pairs (x_A, y_A) and (x_B, y_B) by using a binary mask $\mathbf{M}$. The generation can be described as follows:

$$\hat{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$
$$\hat{y} = \lambda y_A + (1 - \lambda) y_B.$$

The mixing ratio λ is sampled from the Beta distribution, $Beta(\alpha, \alpha)$. In our experiments, we set α to 1 so that the ratio λ is sampled from the uniform distribution $Unif(0, 1)$. When making the binary mask $\mathbf{M}$, we sample a frequency region between (r_y, r_f) whose values are sampled with the following steps:

$$r_y \sim Unif(0, F), \text{ and } r_f = F\lambda \quad (1)$$

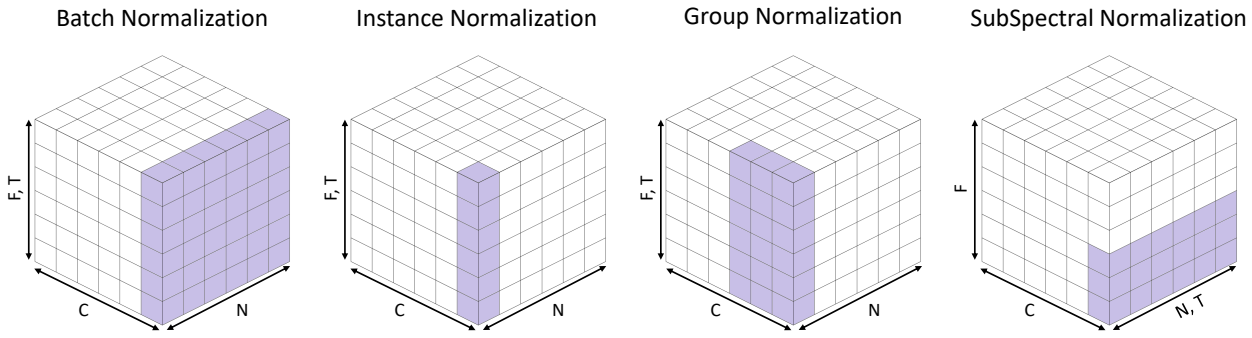

Figure 2: Normalization methods on Frequency-Time audio input, with N as batch axis, C as channels, F as frequency and T as time axis.

Table 2: Number of data for each class of DCASE 2020 Task 1B.

	Dev-Train	Dev-Test
Indoor	2,704	1,297
Outdoor	3,757	1,604
Transportation	2,724	1,284
Total	9,185	4,185

where F is the size of frequency bin of the spectrogram.

4. TRAINING LOSS

The DCASE 2020 Challenge Task 1B evaluates the performance with a macro average accuracy which calculates the accuracy of each class and then performs the average over all class-wise accuracies. With this performance metric, the number of data of each class is not considered: even when the number of test samples is quite different for each class, the class-wise accuracies are equally contributed to the overall average. Table 2 shows the number of data for each category in the development dataset of the DCASE 2020 Task 1B. The outdoor class has much more data than both the indoor and transportation classes. For this data imbalance problem, several methods have been proposed by modifying the training objective [22, 16]. In this work, we apply the class-balanced focal loss [16] which adjusts the data imbalance as

$$\mathrm{CB}_{\mathrm{focal}}(\mathbf{z}, y) = -\frac{1-\beta}{1-\beta^{n_y}} \sum_{i=1}^{C} (1-p_i^t)^{\gamma} \ \log(p_i^t). \qquad (2)$$

Note that we use the *class-balanced* focal loss which replaces α in the original focal loss with $(1-\beta)/(1-\beta^{n_y})$ where $\beta \in [0, 1)$, and n_y is the number of samples.

5. NETWORK COMPRESSION

We consider a network compression method to keep the performance with a reduced model complexity: the model size of task 1B is limited up to 500kB. For this, a network pruning is applied to reduce the number of network parameters, and also a half-precision floating-point is adopted to reduce the number of bits per each parameter.

5.1. Network Pruning

We apply L1-sparsity-based unstructured pruning to reduce the number of network parameters [15]. Using the pruning library included in PyTorch, we perform the global pruning for all convolution weights, and the final model is obtained by an additional fine-tuning of the pruned model. The amount of pruning is determined so that we can have less than 500kB model size which is obtained by the multiplication of the number of parameters and bits per parameter. In the experiments, we obtained the model with 246K parameters by pruning 60% of the original model with 601K parameters. Then, we performed additional fine-tuning of the pruned model with 200 epochs. During the fine-tuning, the learning rate started from 10^{-4} and decayed linearly to 10^{-6}.

6. EXPERIMENTS

6.1. Experimental Setup

We evaluated our proposed architecture using the TAU Urban Acoustic Scenes 2020 3 class dataset which consists of acoustic scene samples recorded in 12 different European cities. Each recording has the audio scene label (one of 10 scenes: e.g. 'airport' or 'shopping mall') and city label (one of 12 cities). For the task 1B, the 10 acoustic scenes are grouped into three major classes such as 'indoor', 'outdoor', and 'transportation'. The development dataset contains 40 hours of data with 14,400 segments. In the experiments, we used 9,185 segments and 4,185 segments for the training and evaluation dataset, respectively: we used the split in the first fold of the validation set. For the evaluation of unseen city, we used 1,440 segments of Milan which are not appeared in the training dataset.

Our ASC framework was implemented using PyTorch, and all experiments were conducted on a GeForce GTX TITAN X GPU with 12Gb RAM. Given a 10 second-long stereo audio input sampled as 48kHz, we extracted 256 logmel features with stereo channel for the input feature of the network. The network was trained by the Adam optimizer with 350 epochs. The learning rates were set to 10^{-4}, exponentially decaying values from 10^{-4} to 10^{-6}, and 10^{-6} respectively for the first 50 epochs, next 200 epochs, and final 100 epochs. For the other experimental setups such as parameter initialization or hyper-parameters, we followed the same setting used in [18].

Table 3: Results on DCASE 2019 Task1A.

Model	Accuracy	#Params
CP-ResNet(ch64) [18]	82.1%	899K
Shared-ResNet(N=3)	82.4%	604K
Shared-ResNet(N=3) + FreqMix	82.7%	604K
Shared-ResNet(N=3) + SSN(S=2)	82.8%	604K
Shared-ResNet(N=3) + FreqMix + SSN(S=2)	**82.9%**	604K

Table 4: Results on DCASE 2020 Task1B. The three values in parentheses on the accuracy column denote the accuracy for indoor, outdoor, and transportation respectively.

Model	Accuracy	#Params	Size
Official Baseline	87.3% (82.0, 88.5, 91.5)	116K	450KB
CP-ResNet(ch64) [18]	97.5% (95.4, 97.5, 99.5)	887K	3,588KB
Ours	97.8% (96.9, 97.7, 98.9)	601K	2,404KB
Ours + CBLoss	**98.0%** (97.8, 97.4, 98.9)	601K	2,404KB
Ours Pruned A	97.9% (97.6, 97.3, 98.9)	246K	491KB
Ours Pruned B	97.8% (96.6, 97.2, 99.5)	246K	491KB
Ours Pruned C	**98.0%** (97.1, 97.1, 99.8)	246K	491KB
Ours Pruned D	97.8% (97.8, 96.4, 99.3)	246K	491KB

6.2. Low-Complexity Acoustic Scene Classification

In this section, we performed the experiment using the DCASE 2019 Task 1A dataset and DCASE 2020 Task 1B dataset to check if the proposed method works well for the low-complex ASC. Given the DCASE 2020 dataset of 3 class labels, it's difficult to check the effectiveness of each component that we proposed since the accuracy is quite high even with the simple official baseline model. Thus, we first evaluated each component using the DCASE 2019 dataset, and then we applied the best setting to the DCASE 2020 Task 1B dataset. Table 3 shows the evaluation results of various models using the DCASE 2019 dataset. With the shared residual structure, we could reduce the model size by more than 30%, and also we obtained a slightly better performance compared to the CP-ResNet. Also, with the FreqMix and SubSpectral Normalization (SSN), we improved the validation accuracy without additional parameters. Combining both FreqMix and SSN, the hights accuracy 82.9% was obtained. Finally, the model, shared-ResNet + FreqMix + SSN, was chosen to evaluate the DCASE 2020 Task 1B dataset.

Table 4 shows the evaluation results of the chosen model using the DCASE2020 Task 1B dataset. Similar with the results in Table 3, our model shows higher performance and has fewer parameters than the baseline CP-ResNet. We obtained more than 10% improvements over the official baseline. However, we obtained less than 1% improvement over the CP-ResNet which shows 97.5% accuracy. As shown in Table tab:dataset, the number of samples is different for each class, and this data imbalance may lead to some performance degradation. To alleviate the imbalance, we applied the class-balanced focal loss (CBLoss) as explained in Sec. 4, and 0.2% performance improvement was obtained as shown in the Table 3.

For this challenge task 1B, we submitted four different models denoted as 'Pruned A', 'Pruned B', 'Pruned C', and 'Pruned D' in Table 4. These four models were obtained based on the 'Ours+CBLoss' model by applying pruning and fine-tuning with different control parameters and initial conditions. We reduced the model size from 2,404KB to 491KB by applying 60% unstructured pruning and half-precision floating-point as explained in Sec. 5. The best model 'Pruned C' shows 98% accuracy with less than 500kB size.

7. CONCLUSION

This technical report describes the details of our approach for low-complexity acoustic scene classification (ASC). There was no complexity limit in the previous ASC challenge, so an ensemble of extensive models could be used. In this report, we consider several techniques such as shared residual architecture, subspectral normalization, and FreqMix. Applying the proposed methods, we achieved 98% accuracy in the official test fold of the development dataset for the DCASE 2020 task1b challenge.

8. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classif. Acoust. Scenes Events*, 2016, pp. 95–99.

[3] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* IEEE, 2005, pp. 158–161.

[4] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," *arXiv preprint arXiv:1807.09840*, 2018.

[6] http://dcase.community/challenge2020/.

[7] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," DCASE2019 Challenge, Tech. Rep., June 2019.

[8] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Acoustic scene classification and audio tagging with receptive-field-regularized CNNs," DCASE2019 Challenge, Tech. Rep., June 2019.

[9] S. Hyeji and P. Jihwan, "Acoustic scene classification using various pre-processed features and convolutional neural networks," DCASE2019 Challenge, Tech. Rep., June 2019.

[10] J. Cho, S. Yun, H. Park, J. Eum, and K. Hwang, "Acoustic scene classification based on a large-margin factorized cnn," *arXiv preprint arXiv:1910.06784*, 2019.

[11] "DCASE2020 task1b description," http://dcase.community/challenge2020/task-acoustic-scene-classification#subtask-b.

[12] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and i-vectors," *Tech. Rep., DCASE2018 Challenge*, 2018.

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

[14] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 825–829.

[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[16] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[17] 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.

[18] K. Koutini, H. Eghbal-Zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[19] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in neural information processing systems*, 2016, pp. 550–558.

[20] Y. N. D. D. L.-P. Hongyi Zhang, Moustapha Cisse, "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[21] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *International Conference on Computer Vision (ICCV)*, 2019.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.