LOW COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH MOFLENET

Technical Report

Oo Yifei

Nanyang Technological University, Singapore
yoo001@e.ntu.edu.sg

Nagisetty Srikanth

Panasonic R&D Center Singapore
srikanth.nagisetty@sg.panasonic.com

ABSTRACT

This technical report details our approach to Task 1: Low-Complexity Acoustic Scene Classification for the DCASE 2024 challenge. We introduced a novel architecture, MofleNet, featuring shuffle channels and residual inverted bottleneck blocks. For the challenge submission, we ensemble this new model with CP-ResNet. To enhance cross-device generalization performance, Freq-MixStyle and Device Impulse Response (DIR) augmentation are applied during training. To meet the constraint of keeping the model size under 128kB, both models are fine-tuned using Quantization Aware Training to perform computations in 8-bit precision. This ensemble method achieved an average accuracy improvement of 6% on the TAU Urban Acoustic Scenes 2022 Mobile development dataset compared to the baseline model of DCASE 2024 Task 1.

Index Terms— MofleNet, CP-ResNet, Ensemble Learning, Quantization Aware Training, Device Impulse Response augmentation, Freq-MixStyle

1. INTRODUCTION

In Task 1 of the DCASE'24 Challenge [1], Low-Complexity Acoustic Scene Classification (ASC), participants are required to design a system that accurately predicts scene labels for 1-second audio clips. In previous years, this task has presented well-known challenges, such as the mismatch between recording devices in the training and test sets, model complexity limits, and energy consumption during training [2, 3]. In addition to strict complexity limits on model size (128 kB) and multiply-accumulate operations (30 million MACs), this year's edition introduces a new constraint: limited availability of labeled data, with only 50%, 25%, 10%, and 5% of the TAU Urban Acoustic Scenes 2022 Mobile development dataset available.

Convolutional Neural Networks (CNNs) are well-established for tackling low-complexity ASC and have dominated the leaderboard in previous DCASE challenges [4, 5, 6, 7, 8]. Lightweight models like MobileNet [5], GhostNet [6], SepNet [7] and blueprint separable convolutions network [8] have been employed to address subsequent challenges. In the DCASE'23 Task 1 challenge, the rank-1 model utilized ensemble of 12 teacher models consisting of 6 Patchout FaSt Spectrogram Transformer (PaSST) variants and 6 variants of CP-ResNet [4] (DCASE'22 Task 1 Rank 1 model) to train a student model CP-mobile [5]. CP-Mobile's performance is heavily dependent on the number of channels in each CPM Block. One of the challenges with CP-Mobile is that reducing the model size often requires sacrificing accuracy. Another issue with CP-Mobile is the MACs involved; scaling the model size down does not proportionately decrease the MACs. This presents a significant challenge in balancing model size, accuracy, and computational efficiency. Additionally, this year's challenge requires participants to train the model on five different sizes of training sets. As noted in [1], training the teacher model on a 100% dataset to distill knowledge into a student model trained on a smaller dataset is not permitted. Therefore, if we use the same approach as the rank 1 submission, we are required to train a total of 60 teacher models. This process is highly resource intensive.

This technical report describes a novel design called MofleNet (MobileShuffleNet), which incorporates combination of channel shuffling and residual inverted bottleneck blocks to the CNN network. This model was efficiently designed to meet the challenge requirements and overcomes the limitations of CP-Mobile. The remainder of the report is structured as follows: Section 2 discuss the data preprocessing and augmentation. Model MofleNet is presented in section 3, Section 4 discuss the ensemble learning by combining MofleNet with CP-ResNet (DCASE'22 Task 1 Rank 1). Training setup, Quantization aware training, results and conclusions are discussed in the subsequent sections.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Preprocessing

For MofleNet, we preprocess the raw 1D time domain audio signals sampled at 32 kHz and convert them to the 2D time-frequency (TF) domain using Short-Time Fourier transform (STFT). This ensures that both the temporal and spectral characteristics of the audio data can be utilized for further analysis and processing. After the frequency domain conversion, we extracted Mel spectrogram corresponding to each audio clip using 256 Mel bands covering the audio frequency bandwidth up to16 kHz. These Mel spectrograms are used as input to MofleNet. For the STFT parameters, a window size of 96 ms with a hop size of 16 ms was considered. For CP-ResNet, the preprocessing settings are identical to MofleNet; however, a hop size of 23.4 ms is used.

2.2. Freq-MixStyle, Device Impulse Response Augmentation and Time Rolling

Frequency MixStyle (FMS) is an approach that aims to leverage frequency-specific information in audio data. It mixes frequency-wise statistics instead of channel-wise statistics in audio processing tasks [9]. FMS normalizes the frequency bands in a spectrogram and then denormalizes them with mixed frequency statistics of two spectrograms. FMS is applied to a batch with a probability specified by the hyperparameter p, and the mixing coefficient is drawn from a Beta distribution parameterized by α.

66 freely available device impulse responses (DIRs) [10] from MicIRP [11] to augment the waveforms are used, like [5]. The characteristic frequency responses of the recording devices in MicIRP make them ideal for simulating a diverse range of recording devices. DIR augmentation is controlled by the hyperparameter p_{DIR}, which defines the probability of convolving a waveform with a DIR.

We set hyperparameters to α=0.3, p_{fms} = 0.4, and p_{DIR} = 0.6 for training MofleNe. For CP-ResNet, we used both FMS and DIR augmentation with hyperparameters α=0.4, p_{fms} =0.8, and p_{DIR}=0.4 according to [5]. In addition, we randomly roll the waveform over time with a maximum shift of 125 ms when training both models.

3. MOFLENET

Inspired by the CP-Mobile [5] and ShuffleNet [12] models, we designed a novel model called MofleNet. One well-known way to reduce model complexity in convolutional neural networks is by using grouped convolution. However, a downside of grouped convolution is that the outputs from certain channels are derived from only a small fraction of input channels, limiting information exchange between these channels. To address this issue, we introduced channel shuffle (refer to Figure 2) after the grouped convolution, which increases the information flow between channel groups and helps in capturing more diverse and comprehensive features. By promoting better mixing of information across different channel groups, it enables richer and more informative feature maps by fully relating the input and output channels [11]. This allows us to reduce the number of parameters without significantly compromising information exchange between channels.

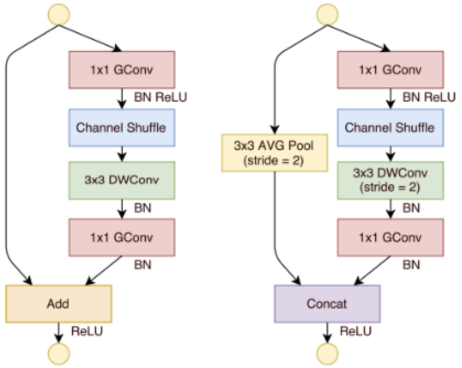

Figure 1: Basic units of ShuffleNet

In MofleNet, we reduced the number of parameters in CP-Mobile [5] by replacing the pointwise expansion convolution of each CPM Block with grouped convolution, setting the number of groups to 2. After this layer, we added a Channel Shuffle layer, followed by depth wise convolution, and pointwise projection convolution, like the CPM Block in [5].

As shown in Fig. 1 [11], the fourth layer in a ShuffleNet unit is a grouped convolution. While this configuration does reduce the number of parameters, our experiments did not demonstrate significant improvement, so the 4th group convolution layer was not considered in our MofleNet design. Figure 3 presents our Mofle blocks.

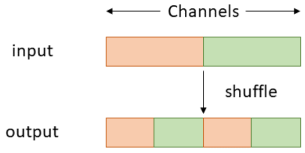

Figure 2: Channe1 Shuffle

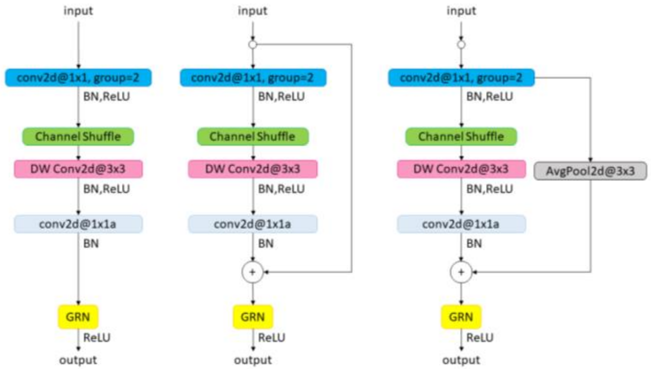

Figure 3: Mofle Blocks: (Left) Transition Block, (middle) Standard Block, (right) Spatial Down sampling Block.

4. ENSEMBLE MODELS

Our submission involves ensemble of two models, MofleNet as mentioned above, and CP-ResNet. To ensure that the models satisfy the challenge constraints of 128 kB and 30 million MACs, we reduced the parameter sizes of MofleNet and CP-ResNet to 59k and 58k respectively, summing up to 117k parameters (equivalent to model size of 117 kB). The MACs for MofleNet and CP-ResNet are 13.4 million (approx.) and 16 million (approx.) summing up to 29.4 million MACs.

4.1. CP-ResNet59

One of the ensemble models is a variant of CP-ResNet, which served as the teacher model for rank-1 model of DCASE'23 Task 1 challenge [5] and achieved rank-1 in the DCASE'22 challenge [4]. Basic blocks of CP-ResNet are presented in Figure 4. We modified the architecture, of the CP-ResNet in [5] to fit the complexity constraints leading to CP-ResNet59. Details of the modifications are as follows:

- Since the number of parameters in the network grows quadratically with its width, we reduced the channel multiplier from 2.0 to 1.4, bringing the parameter count below 64,000.
- Introduced max pooling layers of shape (2x1) and stride (2x1) before the third and fourth (also the last) block to reduce the MACs.

CP-ResNet59 architecture is presented in Table 1.

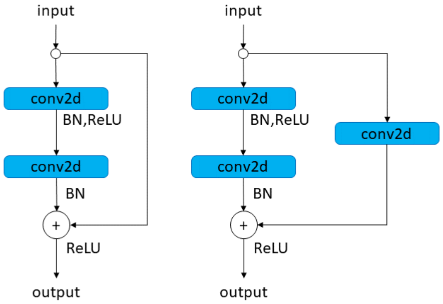

Figure 4: Two Basic Blocks of CP-ResNet

Output Shape Format: Frequency Bands x Time Frames x Number of Channels.
Conv2D@KxK: Conv2D with kernel size KxK.

Table 1: CP-ResNet59 Model Architecture

Operator	Output Shape
Input	256 x 43 x 1
Conv2D@3x3, BN, ReLU	127 x 21 x 32
Max Pool	63 x 10 x 32
Basic Block	63 x 10 x 32
Max Pool	31 x 10 x 32
Basic Block	31 x 10 x 32
Max Pool	15 x 10 x 32
Max Pool (added)	7 x 10 x 32
Basic Block	7 x 10 x 44
Max Pool (added)	7 x 5 x 44
Basic Block	7 x 5 x 26
Conv2D@1x1, BN	7 x 5 x 10
Avg. Pool	1 x 1 x 10

4.2. MofleNet57

The other ensemble model, MofleNet, as presented in Section 3 was further modified to fit the challenge constraint:

- Adjusted the channel multiplier and expansion rate to 1.8 and 2.6, respectively.
- To lower the MACs without impacting the accuracy, we carefully tuned the third Mofle Block of the original CP-Mobile from Block S to Block D with a stride of (2x1) during convolution.

Modified MofleNet architecture (MofleNet57) is presented in Table 2.

Output Shape Format: Frequency Bands x Time Frames x Number of Channels.
Conv2D@KxK: Conv2D with kernel size KxK.
Mofle Block S/D/T: Standard/Spatial Down sampling/transition

Table 2. MofleNet57 Model Architecture

Operator	Stride	Output Shape
Input	-	256 x 64 x 1
Conv2D@3x3, BN, ReLU Conv2D@3x3, BN, ReLU	2x2 2x2	128 x 32 x 8 64 x 16 x 32
Mofle Block S Mofle Block D Mofle Block D	1x1 2x2 2x1	64 x 16 x 32 32 x 8 x 32 16 x 8 x 32
Mofle Block T Mofle Block S	2x1 1x1	8 x 8 x 56 8 x 8 x 56
Mofle Block T	1x1	8 x 8 x 104
Conv2D@1x1, BN Avg. Pool	- -	8 x 8 x 10 1 x 1 x 10

5. TRAINING SETUP

We used PyTorch [13] as the framework to build and train the models. We used the Adam optimizer for training, with a total of 150 epochs and a batch size of 256. The learning rate strategy follows the same approach as in [5], consisting of four phases:

1. **Warmup Phase**: The learning rate exponentially increases for a specified number of epochs until the maximum learning rate.
2. **Constant LR Phase**: The learning rate reaches its maximum value and stays constant.
3. **Linear Decrease Phase**: The learning rate decreases linearly starting from a specified epoch.
4. **Fine tuning Phase**: After the linear decrease phase completes over a specified length, the fine-tuning phase begins, using a learning rate equal to the maximum learning rate multiplied by a specified final value.

The details of the learning rate settings can be found in Table 3. The learning rate of the final phase is obtained by multiplying the “Maximum learning rate” and “Final value of fine-tuning phase” shown in the table.

Table 3. Learning Rate Settings

		MofleNet57	CP-ResNet59
Number of epochs	Phase 1	14	15
	Phase 2	36	35
	Phase 3	84	85
	Phase 4	16	15
Maximum learning rate		0.0009	0.001
Final value of fine-tuning phase		0.005	0.005

6. QUANTIZATION AWARE TRAINING

After completing the training routine outlined above, we fine-tuned our models for 24 epochs using Quantization Aware Training (QAT) [14]. During this fine-tuning phase, we set a peak learning rate of 5×10−5 and linearly decreased it to 10% by epoch 16. We fused all Conv2d + BN + ReLU combinations into a single layer and utilized PyTorch’s 'fbgemm' quantization configuration. All computations were performed in int8, except for those in the GRN layer of MofleNet.

7. RESULTS

The performance of the models across different dataset sizes is shown in Tables 4 and 5. These tables demonstrate that the ensemble of the two models significantly improves accuracy. In Table 4, MofleNet (128k) and CP-ResNet (128k) are models with 128k number of parameters. This table shows that MofleNet generally performs better on larger datasets (100% of the whole dataset [15]) while CP-ResNet performs better on smaller datasets (10%, 5% of the whole dataset [15]). Additionally, our findings suggest that using either MofleNet or CP-ResNet alone without ensembling, results in slight improvement over the baseline model and slightly lower accuracy compared to scaling the sizes down and using the ensemble approach.

Table 4. Model accuracies before QAT

Models	Accuracies				
	100%	50%	25%	10%	5%
Baseline (After Quantization)	0.5699	0.5319	0.5029	0.4529	0.4240
MofleNet-128k	0.6194	0.5868	0.554	0.4910	0.4294
CP-ResNet-128k	0.6065	0.5888	0.5518	0.5082	0.4708
MofleNet57	0.5931	0.5685	0.5246	0.4546	0.4164
CP-ResNet59	0.5957	0.5787	0.5493	0.4928	0.4498
MofleNet57+ CP-ResNet59- v1	0.6255	0.6062	0.5716	0.5123	0.4773
MofleNet57+ CP-ResNet59- v2	**0.6273**	**0.6066**	**0.5744**	**0.5148**	**0.4785**

Table 5. Model accuracies after QAT

Models	Accuracies				
	100%	50%	25%	10%	5%
Baseline	0.5699	0.5319	0.5029	0.4529	0.4240
MofleNet57	0.5879	0.5671	0.5221	0.4540	0.4122
CP-ResNet59	0.5849	0.5752	0.5481	0.4879	0.4492
MofleNet57+ CP-ResNet59- v1	0.6222	0.6004	0.5673	0.5127	**0.4759**
MofleNet57+ CP-ResNet59- v2	**0.6251**	**0.6007**	**0.5718**	**0.5152**	0.4758

In our submission, we applied different weights on the logits of the two models to get the final output logit. The two weight settings are as follows:

v1: CP-ResNet59 – 0.5, MofleNet57 – 0.5
v2: CP-ResNet59 – 0.6, MofleNet57 – 0.4

8. CONCLUSION

In this report, we presented our approach for Task 1: Low-Complexity Acoustic Scene Classification in the DCASE 2024 challenge. We introduced MofleNet, a novel architecture incorporating shuffle channels and residual inverted bottleneck blocks, and used it in an ensemble with CP-ResNet. Our methods included augmentation techniques such as Freq-MixStyle and Device Impulse Response, along with Quantization Aware Training to meet the model size constraint. Our experimental results demonstrated that the ensemble of MofleNet and CP-ResNet significantly improved accuracy compared to individual models by approx. 4% and baseline by approx. 6%. Specifically, MofleNet performed better with larger datasets, while CP-ResNet was more effective with smaller datasets. These findings highlight the importance of model ensembling in addressing the challenges posed by limited labeled data.

9. REFERENCES

[1] Florian Schmid, Paul Primus, Toni Heittola, Annamaria Mesaros, Irene Martín-Morató, Khaled Koutini, and Gerhard Widmer. Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge. 2024. URL: https://arxiv.org/abs/1706.10006.

[2] Irene Martín-Morató, Francesco Paissan, Alberto Ancilotto, Toni Heittola, Annamaria Mesaros, Elisabetta Farella, Alessio Brutti, and Tuomas Virtanen. Low-complexity acoustic scene classification in dcase 2022 challenge. 2022. URL: https://arxiv.org/abs/2206.03835, doi:10.48550/ARXIV.2206.03835.

[3] Irene Martín-Morató, Francesco Paissan, Alberto Ancilotto, Toni Heittola, Annamaria Mesaros, Elisabetta Farella, Alessio Brutti, and Tuomas Virtanen. Low-complexity acoustic scene classification in dcase 2022 challenge. 2022. URL: https://arxiv.org/abs/2206.03835, doi:10.48550/ARXIV.2206.03835.

[4] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer. Knowledge distillation from transformers for lowcomplexity acoustic scene classification. 2022.

[5] Florian Schmid , Tobias Morocutti , Shahed Masoudian , Khaled Koutini , Gerhard Widmer. CP-JKU Submission to DCASE23: Efficient Acoustic Scene Classification with CP-Mobile. 2023.

[6] TaeSoo Kim, Daniel Rho, Gahui Lee, and JaeHan Park. Dual-Strategy Enhancement of Acoustic Scene and Event Classification: Integrating Res2Net, GhostNet, and MobileFormer Architectures. 2023

[7] Yiqiang Cai, Minyu Lin, Chenyang Zhu, Shengchen Li, and Xi Shao. DCASE2023 Task 1 Submission: Device Simulation and Time-Frequency Separable Convolution for Acoustic Scene Classification. 2023

[8] Jiaxin Tan, and Yanxiong Li. Low-Complexity Acoustic Scene Classification Using Blueprint Separable Convolution and Knowledge Distillation. 2023.

[9] Byeonggeun Kim, Seunghan Yang, Jangho Kim, Hyunsin Park, Juntae Lee, and Simyung Chang. Domain Generalization with Relaxed Instance Frequency-wise Normalization for Multi-device Acoustic Scene Classification. 2022. URL: https://arxiv.org/abs/2206.12513v1.

[10] Tobias Morocutti, Florian Schmid, Khaled Koutini, and Gerhard Widmer. Device-Robust Acoustic Scene Classification

via Impulse Response Augmentation. 2023. URL: https://arxiv.org/abs/2305.07499v2.

[11] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, Jian Sun. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. 2017. URL: https://arxiv.org/abs/1707.01083.

[12] Microphone Impulse Response Project. 2017. URL: https://micirp.blogspot.com/?m=1.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, in Advances in Neural Information Processing Systems (NeurIPS). 2019. URL: https://arxiv.org/abs/1912.01703.

[14] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. 2017. URL: https://arxiv.org/abs/1712.05877v1.

[15] Toni Heittola, Annamaria Mesaros, and Tuomas 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), 56–60. 2020. URL: https://arxiv.org/abs/2005.14623.