LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH LIMITED TRAINING DATA

Technical Report

Yun-Fei Shao[1,2]*, Peng Jiang*[1]*, Wei Li*[1]

[1] The School of Mechanical Science and Engineering
Northeast Petroleum University, Daqing 163318, China
shaoyf@tsinghua.edu.cn
jiangpeng@nepu.edu.cn
liweinepu@163.com
[2] Department of Electronic, Tsinghua University, Beijing, 100084, China

ABSTRACT

This report details the architecture we used to address task 1 of the DCASE2024 challenge. In addition to dealing with device mismatches and low complexity limitations, this year's tasks have also added training data limitations. The architecture we propose is based on Mamba, which is a selection state space model with the ability to establish long-term dependencies. Specifically, we designed a variable parallel mamba architecture to further reduce parameters and combined it with the inverted residual block in MobilenetV2 to address training data constraints by adjusting the number of mamba modules and the number of parallels. In addition, we also enhanced the impulse response of audio with energy values greater than the average. Freq-MixStyle (FMS) and audio playback data augmentation methods were used. We apply two model compression schemes: Quantization-Aware Training (QAT), and Knowledge distillation. The proposed system achieves higher classification accuracies than the baseline system. After model compression, our model achieves an average accuracy of 50.1% within the 107.46 K parameters size, 8-bit quantization, and MMACs 16.9 M.

Index Terms— Acoustic scene classification, Mamba, Data augmentation, Model compression, Inverted residual block

1. INTRODUCTION

Acoustic Scene Classification (ASC) [1] is a task of classifying the acoustic scene presented by given audio. It is a multi-class classification task recognizing the recorded environment sounds and specific acoustic scenes that characterize either the location or situation such as park, metro station, tram, etc. Recently, developing signal processing methods to automatically extract audio information has great potential in many application fields, such as searching multimedia based on audio content, manufacturing context-aware mobile devices, and intelligent monitoring systems. Further, we aim for the task1 that become more challenging in comparison with what it was last year on account of stricter restrictions on model size and shorter audio data.

As one of the substantial tasks, acoustic scene classification has been extensively practiced in every challenge. DCASE 2018 and 2019 proposed the mismatch in different recording devices A, B, C, and D. Then in 2020 and 2021, the task of acoustic scene classification was divided into 2 subtasks. Among them, the subtask A worked on the dataset collected with mismatched recording devices in 2020 and added requirements for model complexity in 2021. In 2022 and 2023, the task 1 [2, 3] has no subtasks, but it has stricter restrictions on the complexity of the model than in previous years, such as the model's number of parameters and the multiply-accumulate operations count. Especially, the audio files have a length of 1 second instead of 10 second therefore 10 times more files than in the 2020 version. Added training data limit in 2024 [1].

Over these years, the major network structure that has been adopted is a residual network based on a convolution neural network (CNN) [4, 5, 6]. Nevertheless, the performance of ResNet has decreased since the audio length was shortened to 1s this year. Therefore, this year we have used the Mamba [7] architecture with long-term dependency capabilities.

The rest of the paper is organized as follows. Section 2 presents our systems, including data processing, the proposed student model, the teacher model, and data augmentation. Section 3 introduces the model compression method and experimental setup. Section 4 presents the submitted results. Finally, the conclusion is provided in Section 4.

2. THE SYSTEM

2.1. Data Preprocessing

The one-second audio segments are formatted with a single channel, 32kHz sampling rate, and 16-bit resolution per sample, and in the audio preprocessing stage. we present the spectrum of audio in the log-mel domains with 256 frequency bins.

In our work, we transformed audio data into a power spectrogram by skipping every 500 samples with 3072 length Hanning window. For all models, we randomly roll the waveform over time with a maximum shift of 150 ms, and frequency masking with a maximum size of 48 Mel bins. In addition, we also played back all the training audio data using a laptop computer.

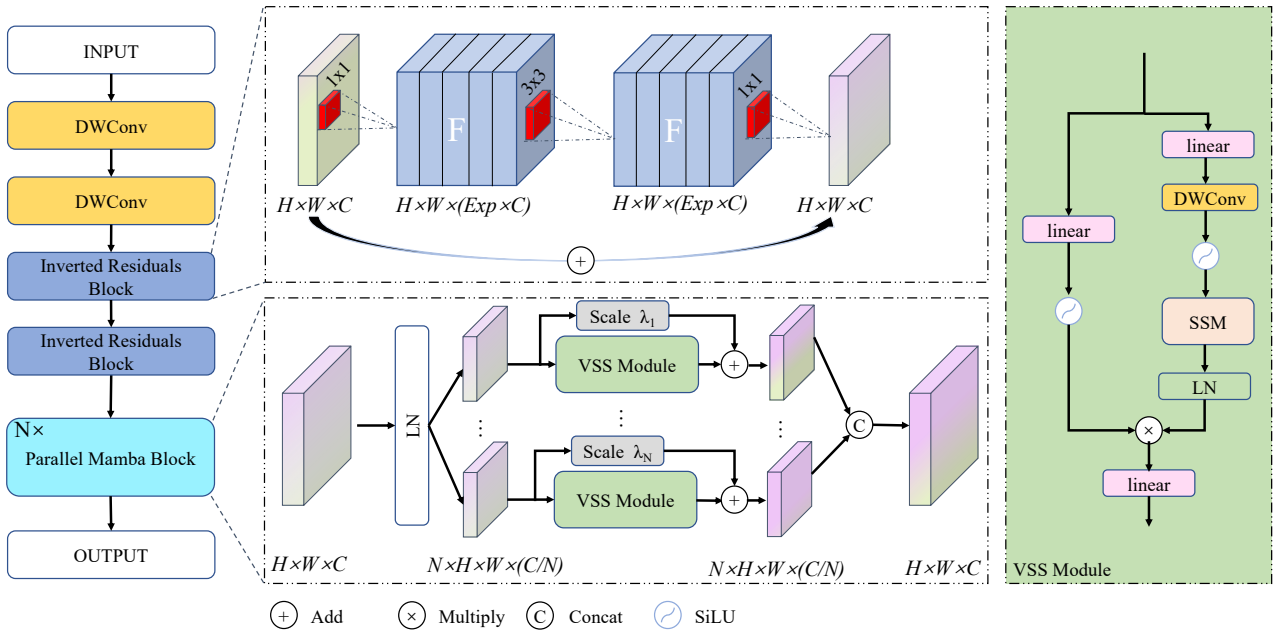

Figure 1: IRMamba is a student network architecture. **DWConv**: Depthwise convolution.

2.2. Data augmentation

2.2.1. *Freq-MixStyle*

Freq-MixStyle (FMS) [8] is a variant of MixStyle [9] that operates on the frequency dimension instead of the channel dimension. It normalizes the frequency dimension in the features and uses mixed frequency statistics for reverse normalization. The specific calculation formula is as follows:

$$\mu_{nf} = \frac{1}{F \cdot T} \sum_{f=1}^{F} \sum_{t=1}^{T} x_{n,c,f,t} \tag{1}$$

$$\sigma_{nf}^2 = \frac{1}{F \cdot T} \sum_{f=1}^{F} \sum_{t=1}^{T} (x_{n,c,f,t} - \mu_f)^2 \tag{2}$$

$$\sigma_{nf} = \sqrt{\sigma_{nf}^2 + \epsilon} \tag{3}$$

where $x \in \mathbb{R}^{N \times C \times F \times T}$. N, C, F, and T denote the batch size, number of channels, frequency dimension, and time dimension, respectively. $\mu_{nf}, \sigma_{nf}^2 \in \mathbb{R}^{N \times F}$ represent the mean and variance. The symbol ϵ is a small value added to σ_{nf}^2 to prevent division by zero.

FMS has two hyperparameters, using probability p_{FMS} = 0.4, and the mixing coefficient is drawn from a Beta distribution parameterized by a hyperparameter α = 0.3.

2.2.2. *Impulse Response*

Tobias et al. [10] and Schmid et al. [8] found that convolution with device impulse response (DIRs) significantly improved the generalization of ASC models to unseen devices. Their approach involves randomly selecting one of 66 DIRs from MicIRP and convolving the waveform with the chosen DIR, trimming trailing samples to maintain the overall length of the waveform. DIR augmentation has a hyperparameter p_{DIR}. it is the probability of convolving the specified waveform with DIR. In addition, we also set an energy threshold parameter p_{thd}, and only waveforms with energy greater than the set threshold will perform impulse response data augmentation with a certain probability p_{DIR}. The hyperparameter is set as follows, with p_{thd} set to 323, which is the average energy of the training set data. p_{DIR} = 0.6. DIR is only used for device A.

2.3. TEACHER MODELS

In our acoustic scene recognition study, we employed knowledge distillation (KD) to enhance the performance of the student model. For the teacher models, we adopted a combination of twelve teacher models based on PaSST and CP-ResNet for model fusion, inspired by the methods described in [8]. To fully exploit the information from all teacher models, we designed a fusion strategy to integrate the logits outputs from the 12 teacher models into a soft target. The fusion formula is as follows:

$$h_{ensemble} = \sum_{k=0}^{12} \alpha_k \cdot \text{logits}_k + \beta \tag{4}$$

where α_k is the weight parameter, with different values for different system models, $\text{logits}_{(k)}$ denotes the logits value of system k. β is the bias parameter, with different values for different categories.

2.4. STUDENT MODEL

In order to address the complexity and contextual information capture issues of models used for acoustic scene classification, we have introduced state space models to address the problem of audio recognition. Firstly, the inverted residual network is used to obtain high-dimensional features, where the application of residual connections helps to address the problem of gradient vanishing/exploding during CNN training [11]. The convolution layer in the inverted residual network uses depthwise separable convolution, which can effectively reduce the number of parameters [12] while ensuring accuracy. Then, temporal context information and spatial correlation information are obtained through variable channel dimensional Parallel Mamba block. The main structure is shown in Fig.1.

The state space models (SSMs) [13, 14, 15], are shown in Fig.1. The space state describes the relationship between input and output, and the space state model (SSM) is used to describe the representation of these states and predict the next state based on the input. The space state represents a sequence $x(t) \in \mathbb{R}$, and through a hidden layer state $h(t) \in \mathbb{R}^N$, the output sequence $y(t) \in \mathbb{R}$ is obtained. This process can be expressed as a linear ordinary differential equation:

$$\begin{cases} h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t) \\ y(t) = \mathbf{C}h(t) \end{cases} \tag{5}$$

where $x(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$, $h(t) \in \mathbb{R}^N$ indicate input signals, output signals, and latent state, respectively. $h'(t)$ represents the derivative of $h(t)$ in time. $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the state transition matrix, $\mathbf{B} \in \mathbb{R}^N$ and $\mathbf{C} \in \mathbb{R}^N$ represents projection parameters matrices. In order to make it more suitable for deep learning, The structural state space model (S4) [13] discretizes this linear system. Specifically, the zero-order hold (ZOH) technique was used to discretize ordinary differential equations (ODEs) in S4 as follows:

$$\begin{cases} \overline{\mathbf{A}} = \exp(\Delta\mathbf{A}) \\ \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \end{cases} \tag{6}$$

where Δ is a timescale parameter, $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are discrete parameters obtained by discretizing $\mathbf{A}$ and $\mathbf{B}$ through timescale parameter Δ. Then, the ODEs of SSMs can be represented as follows:

$$\begin{cases} h'(t) = \overline{\mathbf{A}}h(t) + \overline{\mathbf{B}}x(t) \\ y(t) = \mathbf{C}h(t) \end{cases} \tag{7}$$

For parallel training, the above process can also be replaced by global convolution, defined as:

$$\begin{cases} \overline{\mathbf{K}} = (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}\mathbf{B}}, \ldots, \mathbf{C}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}}) \\ y = x * \overline{\mathbf{K}} \end{cases} \tag{8}$$

where $\overline{\mathbf{K}} \in \mathbb{R}$ denotes the structured convolutional kernel, and L denotes the input sequence length of x.

The specific architecture settings of the student model are shown in Table 1.

Table 1: Student Model Architecture.

Input	Operator	Stride
$256 \times 64 \times 1$	DWConv	2×2
$128 \times 32 \times 8$	DWConv	2×2
$64 \times 16 \times 32$	Inerted Residauls Block	1×1
$64 \times 16 \times 48$	Inerted Residauls Block	2×1
$32 \times 16 \times 48$	Parallel Mamba Block	1×1
$32 \times 16 \times 48$	Max pooling	2×2
$16 \times 8 \times 48$	Parallel Mamba Block	1×1
$16 \times 8 \times 64$	Max pooling	2×2
$8 \times 4 \times 64$	Parallel Mamba Block	1×1
$8 \times 4 \times 72$	Max pooling	2×2
$4 \times 2 \times 72$	Inerted Residauls Block	1×1
$4 \times 2 \times 128$	Conv2D@1×1	
$4 \times 2 \times 10$	Avg pooling	

3. KNOWLEDGE DISTILLATION AND QUANTIZATION TRAINING

Knowledge distillation (KD) has been demonstrated to effectively compress models and enhance the generalization ability of smaller models in ASC tasks. The loss function consists of two components: the label loss and the knowledge distillation loss.

The label loss, a classification loss, quantifies the disparity between the model's predictions and the ground truth labels by calculating the cross-entropy loss (CELoss). Knowledge distillation loss measures the discrepancy between the outputs of the teacher and student models using Kullback Leibler divergence (KDloss). Its objective is to align the student model's output with that of the teacher model. The total loss can be represented as:

$$\text{Loss} = \lambda\text{CELoss} + (1-\lambda)\tau^2\text{KDLoss} \tag{9}$$

where, λ is a weight-balancing hard label loss and distillation loss, updated during training. τ is the KD temperature.

After completing the knowledge distillation training, we fine-tune the student models using Quantization Aware Training (QAT) [16] with the observer set to "fbgemm" to convert all parameters and computations involved in the student models to int8. In the forward pass, we perform all computations in int8 except for the computations performed in the LayerNorm layer and SSM block.

3.1. Experimental Setup

We train our models for 150 epochs on GPU, and the batch size is set to 256. We use the AdamW optimizer with a cosine schedule with the warmup. Warmup steps are set to 2000. the peak value of the learning rate is 0.005. For KD, τ = 2, λ = 0.02.

4. SUBMISSIONS

Due to increased training data limitations in this year's competition tasks, we are trying to find the best model that can adapt to different training data volumes. As shown in Table 2, there are three systems we have submitted, named S1, S2, and S3. We explored different numbers of Parallel Mamba Blocks and different numbers of VSS modules within the constraints of model complexity and MMACs. In addition, we also explored the impact of replay data on model accuracy, where the benefit is minimized when the training data approaches 100%.

Table 2: Model configurations submitted to the challenge of DCASE 2024. **PMBN**: the number of the Parallel Mamba block. **VSSN**: the number of the VSS Module

ID	PMBN	VSSN	#Param.(K)	MMACs	5% Acc.	10% Acc.	25% Acc.	50% Acc.	100% Acc.
S1	3	6	107.46	16.9	51.06	53.52	59.67	62.3	64.62
S2	5	17	121.9	17.3	48.73	53.09	59.51	62.12	65.1
S3	4	185	126.41	16.8	46.57	51.45	58.34	62.25	65.08

5. CONCLUSION

In this report, we present the methods and techniques we used in task 1 of the DCASE2024 challenge. We have introduced state space models for the first time to process acoustic signals. It has excellent long-sequence processing capabilities and linear complexity, making it more suitable for task scenarios. Compared to the baseline system, the accuracy has been improved by nearly 10%. The main improvement is attributable to an efficient, novel architecture, Parallel Mamba, constructed of SSM. Additionally, we also used data augmentation methods such as impulse response, FMS, and audio playback, to further improve the recognition performance of the model.

6. REFERENCES

[1] http://dcase.community/challenge2024/.

[2] https://dcase.community/challenge2022/ task-low-complexity-acoustic-scene-classification.

[3] https://dcase.community/challenge2023/ task-low-complexity-acoustic-scene-classification.

[4] 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," *arXiv preprint*, 2022.

[5] H. Hee-Soo, J. Jee-weon, S. Hye-jin, and L. Bong-Jin, "Clova submission for the DCASE 2021 challenge: Acoustic scene classification using light architectures and device augmentation," DCASE2021 Challenge, Tech. Rep., June 2021.

[6] L. Byttebier, B. Desplanques, J. Thienpondt, S. Song, K. Demuynck, and N. Madhu, "Small-footprint acoustic scene classification through 8-bit quantization-aware training and pruning of ResNet models," DCASE2021 Challenge, Tech. Rep., June 2021.

[7] R. Wu, Y. Liu, P. Liang, and Q. Chang, "Ultralight vm-unet: Parallel vision mamba significantly reduces parameters for skin lesion segmentation," *arXiv preprint*, 2024.

[8] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase23: Efficient acoustic scene classification with cp-mobile," DCASE2023 Challenge, Tech. Rep., May 2023.

[9] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint*, 2021.

[10] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 176–180.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake, USA, 2018, pp. 4510–4520.

[12] 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*, 2017.

[13] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint*, 2022.

[14] J. Ruan and S. Xiang, "Vm-unet: Vision mamba unet for medical image segmentation," *arXiv preprint*, 2024.

[15] G. Wang, X. Zhang, Z. Peng, T. Zhang, X. Jia, and L. Jiao, "S^2mamba: A spatial-spectral state space model for hyperspectral image classification," *arXiv preprint*, 2024.

[16] 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," *arXiv preprint*, 2017.