Low-Complexity Acoustic Scene Classification Base on Depthwise separable CNN

Technical Report

Zhicong Liang, Pengyuan Xie, Zhe Wang, Wenbo Cai

NetEase
Guangzhou, China

ABSTRACT

This report outlines our submission for DCASE2023 Task1, which focuses on Low Complexity Acoustic Scene Classification. To meet this requirement, we implemented the Depthwise Separable CNN method to construct our model. This approach significantly reduces model size while improving accuracy. Additionally, we applied SpecAugment and mixup as data augmentation techniques. To further enhance our model's performance, we employed Knowledge Distillation, teaching the submission model from larger models. Overall, these techniques enable us to achieve better results on the task.

Index Terms— Depthwise Separable CNN, SpecAugment, Mixup, Knowledge Distillation

1. INTRODUCTION

The task1 of the DCASE2023 challenge is to classify a test recording into one of the predefined ten acoustic scene classes [1]. This targets acoustic scene classification with devices with low computational and memory allowance, which impose certain limits on the model complexity, such as the model’s number of parameters and the multiply-accumulate operations count. In addition to low-complexity, the aim is generalization across a number of different devices. For this purpose, the task will use audio data recorded and simulated with a variety of devices.

The Depthwise Separable CNN [6][7] is a type of Convolutional Neural Network architecture that reduces the computational complexity of traditional CNNs. Due to its efficiency and effectiveness, it has demonstrated promising results in a variety of applications, making it an ideal choice for the current competition's requirements. Additionally, we utilized Knowledge Distillation [9], a technique in machine learning that enables the transfer of knowledge from a large, complex model (known as the teacher model) to a smaller, simpler model (known as the student model). By integrating these techniques, we were able to create a powerful and efficient model for the task.

2. FEATURE EXTRACTION AND DATA AUGMENTATION

2.1. Reassembling Audios

For this task, we utilized the TAU Urban Acoustic Scenes 2022 Mobile development dataset [2], which contains the same content as the TAU Urban Acoustic Scenes 2020 Mobile development dataset but with audio files that are only 1 second in length. As a result, there are ten times more files in the 2022 version. To create a more informative dataset, we reassembled the 10-second dataset from the 1-second dataset. During model training, we randomly cropped the 10-second audio into 1-second audio segments as part of our feature extraction process. This approach enabled us to extract relevant acoustic features from the audio dataset, ultimately leading to improved model performance.

2.2. Feature Extraction

To extract relevant features from the 1-second audio with a sampling rate of 44.1kHz, we utilized Log-mel Energies Features [5] as the input for our model. We employed the Short Time Fourier Transform with a hamming window size of 3528 and overlap of 25% to extract these features. Additionally, we used 4096 FFT points and 256 log-mel filter banks to enhance the quality of our feature extraction process. As a result, the shape of our model's input feature was 1 x 256 x 51, which adequately captured the relevant acoustic attributes of the audio dataset.

2.3. SpecAugment and Mixup

To enhance the quality and robustness of our model, we employed the SpecAugment [3] technique, which comprises two primary operations: frequency masking and time masking. In the frequency masking operation, we randomly masked out frequency bands in the spectrogram by setting them to zero, simulating missing data or background noise. Similarly, in the time masking operation, we randomly masked out contiguous time segments in the spectrogram by setting them to zero, further simulating missing data. In training, the size of both frequency and time masking was randomly chosen to be between 0 and 10. By applying these masking operations in both the frequency and time domains, we were able to increase the diversity of our training data and improve our model's ability to generalize to new and unseen data.

Mixup [4] is a widely used data augmentation technique in classification models. This technique involves taking two examples with their corresponding labels and generating a new example and label by randomly combining the two examples and their labels using a weighted sum. The weight of each example is determined by a random value lambda, which is sampled from a beta distribution with parameters (0.5, 0.5) in training. With mixup, the resulting augmented training set is a combination of original and synthetic data that better captures the underlying distribution of the data, enhancing the model's ability to generalize to new data.

3. ARCHITECTURES

3.1. Model Architecture

The model architecture is based on a Convolutional Neural Network (CNN). As shown in Table 1, the model consists of several parts, including Input, DW1, DW2, DW3, and Output. The Input layer contains a Convolution layer with a 7x7 kernel. The DW1 layer contains a DwBlock and a MaxPooling layer with a 5x5 kernel. The DW2 layer starts with a Convolution layer with a 5x5 kernel, followed by a DwBlock. The DW3 layer is similar to DW1 but with a MaxPooling layer with a kernel size of 10 x 4. Finally, the Output layer consists of a flatten layer and 2 Linear Layers. When the hidden size is set to 18, the number of parameters is 31.26K (maximum is 32K when using float32), and the number of MMAC (million multiply-accumulate operations) is 29.6 (maximum is 30).

	Architecture	Parameters	Output Shape
Input	Conv	900	(18, 256, 51)
DW1	DwBlock	1044	(18, 256, 51)
	MaxPool2d	-	(18, 51, 10)
DW2	Conv	8118	(18, 51, 10)
	DwBlock	1044	(18, 51, 10)
DW3	DwBlock	1044	(18, 51, 10)
	MaxPool2d	-	(18, 5, 2)
Output	Flatten	-	180,
	Linear	18100	100,
	Linear	1010	10,

Table 1: Model Architecture

3.2. DwBlock Architecture

As illustrated in Figure 1, the DwBlock architecture comprises four main components: BatchNorm [11], ReLU activation, Depthwise Convolution, Pointwise Convolution, and a Residual Connection [8] that adds the input features to the output features. The Depthwise Convolution [12] operation is a computationally efficient variant of the standard convolution operation, which reduces the number of parameters and computations required by the network. This operation applies a single filter to each input channel separately, rather than using a different filter for each combination of input and output channels. The Residual Connection helps to mitigate the vanishing gradient problem by allowing the gradient to be propagated more easily through the network. This connection adds the input features to the output features, thereby creating a shortcut path for gradient flow. The BatchNorm and ReLU activation functions are commonly used to normalize and scale the output of the Depthwise Convolution operation and introduce nonlinearity into the network.

3.3. Knowledge Distillation

The Knowledge Distillation [9] process involves using a fused teacher model to distill knowledge into a student model. Four larger teacher models are trained with different configurations, each having the same architecture as the student model but with a much larger size of 2.3M parameters and hidden size set to 256. The ensemble method of averaging is then applied to fuse the outputs of the four teacher models and create a better fused model.

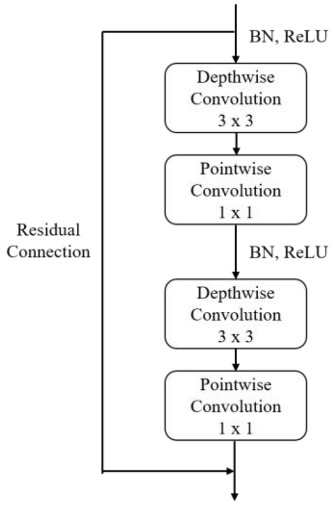
BN, ReLU
Depthwise Convolution 3 x 3
Pointwise Convolution 1 x 1
Residual Connection
BN, ReLU
Depthwise Convolution 3 x 3
Pointwise Convolution 1 x 1

Figure 1: The architecture of Dwblock

The soft targets are generated by the fused teacher model, which are used to train the student model using a weighted sum of soft loss and hard loss. The soft loss measures the difference between the student model's predicted probabilities and the probabilities generated by the fused teacher model, while the hard loss measures the difference between the student model's predicted labels and the true labels.

4. EXPERIMENT

To train and test our model, we used the official setup of the development dataset. Initially, we trained four different teacher models for 200 epochs using the Adam optimizer with a batch size of 64. The learning rate started from 0.01 and gradually decreased as the epoch increased. We then applied the knowledge distillation method to train the student model. We used a temperature of 2 and set the weighting of the soft loss to 0.5. In the KD framework, we used the Adam [10] optimizer with the same configurations as teacher training, but we increased the training epoch to 600. Table 2 shows the performance of our model, which achieved higher accuracy compared to the baseline model. The training parameters such as learning rate, batch size, and number of epochs were carefully tuned to optimize the accuracy of the model.

Model	Parameters	Accuracy
Baseline	46,512	42.9%
Teacher1	2,318,678	58.7%
Teacher2	2,318,678	59.0%
Teacher3	2,318,678	58.9%
Teacher4	2,318,678	58.6%
Student w/o KD	31,260	51.3%
Student (submission)	31,260	54.9%

Table 2: Results for development dataset

5. CONCLUSION

In this report, we present our model for the task of Low-Complexity Acoustic Scene Classification. Our model takes log-mel energies as input and incorporates data augmentation techniques such as SpecAugment and mixup to enhance its robustness. To reduce the complexity of our model, we use a Depthwise separable CNN architecture. Additionally, we employ the knowledge distillation technique to transfer knowledge from larger teacher models to our smaller student model, which leads to improved performance. Overall, our approach shows promising results in achieving high accuracy while maintaining low complexity.

6. REFERENCES

[1] Irene Mart ín-Morat ó, 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.

[2] 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.

[3] Park, Daniel S., William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.” In Interspeech 2019.

[4] H. Zhang, M. Ciss é, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyondempiricalriskminimization,”in6thInternational Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[5] 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.

[6] 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.” 2017.

[7] F. Chollet, “Xception: Deep learning with depthwise separable convolutions.” 2016.

[8] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition.” Cornell University - arXiv.

[9] G. E. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” CoRR, vol. abs/1503.02531, 2015.

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

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

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Commun. ACM, p. 84–90, may 2017.