LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH MISMATCH-DEVICES USING SEPARABLE CONVOLUTIONS AND COORDINATE ATTENTION

Technical Report

Yifei Xin[1], *Yuexian Zou*[1,*], *Fan Cui*[2], *Yujun Wang*[2]

[1]School of ECE, Peking University, Shenzhen, China
[2]Xiaomi Corporation, Beijing, China

ABSTRACT

This report details the architecture we used to address Task 1 of the of DCASE2022 challenge. Our architecture is based on 4 layer convolutional neural network taking as input a log-mel spectrogram. The complexity of this network is controlled by using separable convolutions in the channel, time and frequency dimensions. Moreover, we introduce a novel attention mechanism by embedding positional information into channel attention, which we call coordinate attention to improve the accuracy of a CNN-based framework. Besides, we use SpecAugment++, time shifting and test time augmentations to further improve the performance of the system.

Index Terms— Acoustic Scene Classification, Separable Convolutions, Coordinate Attention, Data Augmentation

1. INTRODUCTION

Extracting information from audio signals can be a great improvement in existing applications or future products (home assistants, wildlife monitoring, autonomous cars, etc.). Machine listening is understood as the set of algorithms that are capable of extracting relevant information from audio. One of the most common tasks in this field is known as Acoustic Scene Classification (ASC) [1–4]. The ultimate goal is to extract context information from the audio, more specifically, to predict the location where the audio is produced (park, metro station, airport, etc.). This problem has been addressed in all previous editions of DCASE, and has been modified with different restrictions [5]. In this report, an ASC system is designed to be limited by the size of the model and the extra difficulty that the audios used in the training come from different audio sources (mismatch devices).

2. MODEL

2.1. Separable Convolutions

Our architecture is based on CNN6 described in [6]. This network consists of 4 convolutional layers using 5 × 5 filters, followed by a global pooling layer and a final MLP.

To reduce the number of parameters, we replace each of the original 5 × 5 convolutional layers with separable convolutions along the channel, time and frequency axes [7].

Let $X \in \mathbb{R}^{n \times c}$ denote a feature map of spatial dimension n with c_{in} channels. In the original convnet CNN6, the feature maps from one layer to the next satisfy

$$Y = f_{5\times5}(X), \tag{1}$$

where $f_{5\times5} : \mathbb{R}^{n\times cin} \rightarrow \mathbb{R}^{n\times cout}$ denotes a regular convolution layer with a kernel of size 5 × 5 giving a feature map Y of spatial dimension n with c_{out} channels. To reduce the parameter numbers, we replace (1) by

$$Y = h_{3\times1}(g_{1\times1}(X)) + h_{1\times3}(g_{1\times1}(X)), \tag{2}$$

where $g_{1\times1} : \mathbb{R}^{n\times cin} \rightarrow \mathbb{R}^{n\times cout}$ denotes a regular convolution layer with a kernel of size 1 × 1, and $h_{3\times1}, h_{1\times3} : \mathbb{R}^{n\times cout} \rightarrow \mathbb{R}^{n\times cout}$ denote two different channel-wise convolutions with kernels of size 3 × 1 and 1 × 3, respectively, dilated by 2 to keep a receptive field similar to the one of $g_{5\times5}$.

2.2. Coordinate Attention

Recent studies on mobile network [8] design have demonstrated the remarkable effectiveness of channel attention (e.g., the Squeeze-and-Excitation attention) [9] for lifting model performance, but they generally neglect the positional information, which is important for generating spatially selective attention maps. In this paper, we introduce a novel attention mechanism by embedding positional information into channel attention, which we call coordinate attention [10]. Unlike channel attention that transforms a feature tensor to a single feature vector via 2D global pooling, the coordinate attention factorizes channel attention into two 1D feature encoding processes that aggregate features along the two spatial directions, respectively. In this way, long-range dependencies can be captured along one spatial direction and meanwhile precise positional information can be preserved along the other spatial direction. The resulting feature maps are then encoded separately into a pair of direction-aware and position-sensitive attention maps that can be complementarily applied to the input feature map to augment the representations of the objects of interest. The coordinate attention is simple and can be flexibly plugged into CNN-based models.

A coordinate attention block can be viewed as a computational unit that aims to enhance the expressive power of the learned features for CNN networks. It can take any intermediate feature tensor X as input and outputs a transformed tensor with augmented representations of the same size to X. The more detailed implementation of this block can be seen in [10].

*Corresponding author

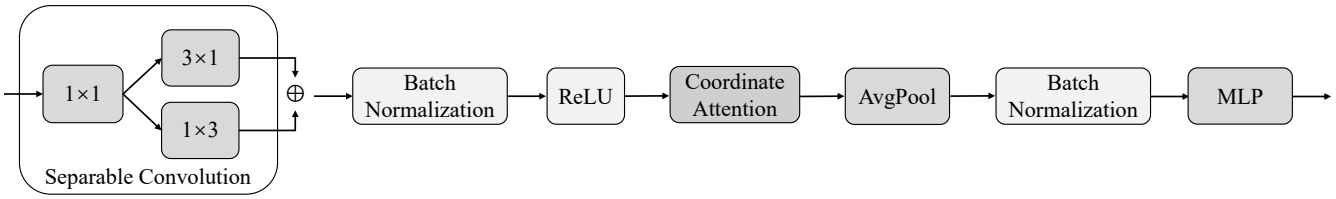

Figure 1: Network architecture. The separable convolution block shows the architecture we use in place of 5 × 5 convolutions in CNN6.

2.3. Complete Architecture

The complete architecture is presented in Figure 1. The layers consist of a convolutional layer, two batch normalisation layers, a ReLU activation, a coordinate attention block, an average pooling layer and a MLP network. The average pooling layer reduces the resolution by 2 in the time and frequency axes with a kernel of size 2 × 2. The global pooling layer consists of (i) a global averaging pooling along the frequency axis followed by (ii) a global averaging and a max pooling along the time axis, the results of both pooling being summed together to yield a feature vector of size 128 that enters the final MLP for classification. This MLP contains two layers with a hidden dimension of 128 using a ReLU activation in the hidden layer.

3. DATA AUGMENTATIONS

3.1. SpecAugment++

Different from other popular data augmentation methods such as SpecAugment [11] and mixup [12] that only work on the input space, SpecAugment++ [13] is applied to both the input space and the hidden space of the deep neural networks to enhance the input and the intermediate feature representations. For an intermediate hidden state, the augmentation techniques consist of masking blocks of frequency channels and masking blocks of time frames, which improves generalization by enabling a model to attend not only to the most discriminative parts of the feature, but also the entire parts.

Let $x \in \mathbb{R}^{T \times F}$ denote the intermediate hidden state (or the input spectrogram), where T and F denote the number of frames and frequency bins, respectively. Time masking is applied so that t consecutive time frames $[t_0, t_0 + t]$ are masked, where t is chosen from a uniform distribution from 0 to the time mask parameter t', and t_0 is chosen from $[0, T - t]$. Similarly, frequency masking is applied so that f consecutive frequency bins $[f_0, f_0 + f]$ are masked, where f is first chosen from a uniform distribution from 0 to the frequency mask parameter f', and f_0 is chosen from $[0, F - f]$.

For masking schemes, we use the mini-batch based mixture masking (MM). MM utilizes the time frames and frequency channels from another sample for masking, which mixes the masking regions of the hidden states of the two samples by the mean. To explain, if the hidden state in the l-th layer of the target sample is to be augmented, we randomly select another sample within the same mini-batch as the target sample and use the hidden state in the l-th layer of the selected sample for masking.

3.2. Time Shifting

The goal of time shifting [14] is to encourage the model to learn coherent predictions. Effectively, given a clip of multiple audio frames $X = [x_1, ..., x_T]$, $x \in \mathbb{R}^{T \times F}$, time rolling of length η will shift (and wrap around) the entire sequence by η frames to $X_0 = [x_\eta, ..., x_T, ..., x_1, ..., x_{\eta-1}]$. For each audio-clip, we draw η from a normal distribution $N(0, 10)$, meaning that we randomly either shift the audio clip forward or backward by η frames.

Table 1: Our results compared with the challenge baseline.

Method	Model size (KB)	Log Loss	Accuracy
Challenge Baseline	46.512	1.575	0.429
Ours	75.562	1.295	0.603

3.3. Test Time Augmentation

Test-time augmentation (TTA) is commonly used in image classification in order to increase the accuracy of a model predictions [15]. Contrary to data augmentations during training, TTA is applied during inference. The main idea of TTA is that by making several randomly augmented copies of the input sample, then averaging the outputs for the augmented samples, more accurate predictions can be made without changing the model. In our experiments, We average the softmax predictions from 30 different augmentations.

4. EXPERIMENTS

4.1. Training Details

The model is trained for 150 epochs, with a batch size of 32, a weight decay of 10^{-5}, using AdamW with a starting learning rate of 10^{-3} and a cosine annealing scheduler decreasing the learning rate to 10^{-5}. We follow the same training pipeline in [6] to train our model and evaluate it on the validation set. We use two dropout layers [16]: the first on the input of the final MLP and the second on its hidden layers. These layers drop each neuron with probability 0.1. All systems are trained by applying a softmax on the final logits and using the cross-entropy loss.

4.2. Results

The overall results obtained by our system can be seen in Table 1. Our model achieves the best overall performance of 60.3% which improves DCASE baseline by 17.4%.

5. CONCLUSION

This technical report aims to describe our low-complexity ASC models for DCASE 2022 task 1. We use separable convolutions with the coordinate attention block as our network. Besides, we also use SpecAugment++, time shifting and test time augmentations to further improve the performance of our system. Our experiments

conducted on DCASE 2022 Task 1 Development dataset have fulfilled the requirement of low-complexity and achieved a log-loss of 1.295 and an accuracy of 60.3%, improving DCASE baseline by 17.4% within the 128 KB model size.

6. 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 *DCASE*, 2016, pp. 95–99.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification: an overview of dcase 2017 challenge entries," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 411–415.

[4] L. Yang, X. Chen, and L. Tao, "Acoustic scene classification using multi-scale features." in *DCASE*, 2018, pp. 29–33.

[5] J. Naranjo-Alcazar, S. Perez-Castanos, M. Cobos, F. J. Ferri, and P. Zuccarello, "Task 1a dcase 2021: Acoustic scene classification with mismatch-devices using squeeze-excitation technique and low-complexity constraint," *arXiv preprint arXiv:2107.14658*, 2021.

[6] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[7] A. Bursuc, G. Puy, and H. Jain, "Separable convolutions and test-time augmentations for low-complexity and calibrated acoustic scene classification," 2021.

[8] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," in *European Conference on Computer Vision*. Springer, 2020, pp. 680–697.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[10] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 713–13 722.

[11] 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.

[12] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, "How does mixup help with robustness and generalization?" *arXiv preprint arXiv:2010.04819*, 2020.

[13] H. Wang, Y. Zou, and W. Wang, "Specaugment++: A hidden space data augmentation method for acoustic scene classification," *arXiv preprint arXiv:2103.16858*, 2021.

[14] H. Dinkel, M. Wu, and K. Yu, "Towards duration robust weakly supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[15] I. Kandel and M. Castelli, "Improving convolutional neural networks performance for image classification using test time augmentation: a case study using mura dataset," *Health information science and systems*, vol. 9, no. 1, pp. 1–22, 2021.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.