# LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE DEVICES

Technical Report

*YiHao Chen[1], YuFeng Ma[1], Min Liu[1], Liang He[2], Ying Hu[3], MinQiang Xu[1]*

[1] SpeakIn Technology
[2] Tsinghua University,Department of Electronic Engineering
[3] Xinjiang University, School of information science and engineering

## ABSTRACT

This report describes our submission to the Task1 Acoustic Scene Classification in the Dcase 2021 challange. Final submission includs 4 results based on ResNet and SEResNet architectures. We perform several analysis of different backbones and also do experiments to confirm whether the pooling layer is needed. Due to the lack of training data, we try a variety of data enhancement methods including specaugment[1], cutout[2], audio acceleration and deceleration. To meet the requirement of model size, we also do pruning to the models.

***Index Terms*—** Acoustic Scene Classification, Dcase 2021 challange

## 1. INTRODUCTION

As mentioned in the abstract, this report describes the AIAL-XJU Team submissions for the Dcase 2021 task1 subtask A challenge. Sounds carry a large amount of information about our everyday environment and physical events that take place in it. The Dcase challenge is intended to perceive the sound scene we are within(busy street, office, etc.), and recognize individual sound sources(car passing by, footsteps, etc.). This challenge has several tasks including Acoustic scene classification, Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions, Sound Event Localization and Detection with Directional Interference and so on. Here we only attends the first task of Acoustic scene classification, Low-Complexity Acoustic Scene Classification with Multiple Devices. Since a model complexity limit of 128KB is set for the non-zero parameters, we spend more energy on model miniaturization.

The rest of this report is organized as follows. Section 2 gives several components of our systems such as model architectures, metric functions. In Section 3, the setup of our experiments are presented. Section 4 concludes this report.

## 2. SYSTEM DESCRIPTION

Considering that the training data is less, we decided to increase the sample with variable speed data. At the same time, SpecAugment is applied directly to the feature inputs of a neural network. The augmentation policy consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps. In the choice of backbone, we decided to use resnet and seresnet. Finally, in order to fulfill the parameter requirements of the challenge, we decided to use pruning technology to remove redundant parameters.

Table 1: ResNet18 backbone. The input shape is H × W as H represents the feature dimension and W represents the frame length. Here S stands for stride, K stands for kernel size and C is number of channels.

| **layer** | **parameter** | **output** |
|---|---|---|
| stem | **C** = 8, **K** = 3, **S** = 1 | 8 × H × W |
| Res1 | **C** = 8, **K** = 3, **S** = 1 | 8 × H × W |
| Res2 | **C** = 16, **K** = 3, **S** = 1 | 16 × H × W |
| Res3 | **C** = 16, **K** = 3, **S** = 1 | 16 × H × W |
| Res4 | **C** = 32, **K** = 3, **S** = 1 | 32 × H × W |

### 2.1. Bckbone

#### *2.1.1. ResNet backbone*

This backbone is based on ResNe18[6] topology. This network takes 2-dimensional features as input and the stem layer is composed of one convolution layer with kernel size as 3 and stride as 1, one batch normalization layer and ReLU. The following residual layers are as same as the original structure. The whole architecture is as Table.1

#### *2.1.2. Se-ResNet backbone*

The SEResNet18 backbone is just as the same structure as ResNet18 except that the SEBasicblock[3] has one more additional Squeeze-and-excitation layer[8] than Basicblock. Since the residual function of Basicblock can be formulated as:

$$y = F(x, W_i) + D(x) \tag{1}$$

x and y are input and output of the layer, and the function F here are convolution layers with batch normalization and ReLU operations and Wi represent the weights of the i-th layer. The D function here is a downsampling function which is a convolution with kernel size as 1 to match the shape of feature maps. Adding SE after the residual part turn this function as

$$y = SE(F(x, W_i)) + D(x) \tag{2}$$

### 2.2. Metric

In this challenge, we use two kinds of loss functions in the experiments. Softmax loss with label smooth and CM-softmax loss which is the combination of Additive Margin softmax loss(AM) and Additive Angular Margin softmax loss. The scale of CM is fixed as 30 and aam margin set to be 0.2, am margin set to be 0.1.

### 2.3. Pruning

We use network slimming, which is a simple effective network training scheme, to reduce the number of parameters[4]. We impose L1 regularization on the scaling factors in batch normalization(BN) layers. Pushing the values of BN scaling factors towards zero with L1 regularization enables us to identify insignificant channels. Then we obtain a model in which many scaling factors are near zeros.

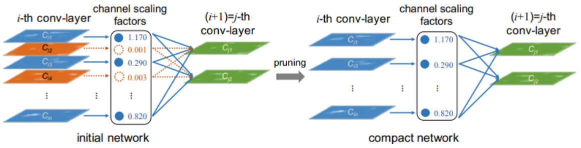

Figure 1: The prune figure.

Then we can prune channels by removing all their incoming and outgoing connections and corresponding weights. The sparse model will be degraded in capacity, so it needs to be finetuned. We use the same optimization setting as in training.

## 3. EXPERIMENT SETUP

### 3.1. Dataset

The development dataset for this task is TAU Urban Acoustic Scenes 2020 Mobile, development dataset[5]. The dataset contains recordings from 12 European cities in 10 different acoustic scenes using 4 different devices. Additionally, synthetic data for 11 mobile devices was created based on the original recordings. The dataset is provided with a training/test split in which 70% of the data for each device is included for training, 30% for testing.

### 3.2. Features

• log-mel energies: 40 dimensional log-mel features(including energy) The feature has frame-lengths of 40ms with 50% hop size.

### 3.3. Data Enhancement

We used 0.9x speed and 1.1x speed audio for data enhancement and later used SpecAugment, which is applied directly to the feature. The augmentation policy consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps.

### 3.4. Experiments Results

From table 2, it can be conclued that using specaugment will reduce the performance on the verification set, which is predictable, but at the same time it will enhance the generalization ability of the model. Using specaugment and speed transformation at the same time can enrich the data set and improve the robustness while maintaining the correct rate of verification. From Table 3, it can be concluded that the classification performance of the pruned or fine-tuned models degrade only when the pruning ratio surpasses a threshold. The fine-tuning process can typically compensate the possible accuracy loss caused by pruning. Only when the pruning ratio goes beyond 0.7, the test error of fine-tuned model falls behind the baseline model.

Table 2: The impact of specaugment and spx3

| **model** | **num_parameters** | **valid_acc** |
|---|---|---|
| seresnet18 | 107k | 77.4% |
| specaug_seresnet18 | 107k | 73.8% |
| spx3_specaug_seresnet18 | 107k | 75.3% |

Table 3: Pruning effect comparison

| **model** | **num_parameters** | **valid_acc** |
|---|---|---|
| seresnet18 | 107k | 75.3% |
| (**prune_ratio 0.6**)seresnet18 | 73k | 76% |
| (**prune_ratio 0.65**)seresnet18 | 63k | 76.1% |
| (**prune_ratio 0.7**)seresnet18 | 42k | 69.4 |

## 4. CONCLUSION

In this technical report, we try to get more training data through data enhancement. We also explore different model architecture to get better performance. In order to meet the requirements of model size, we do pruning to the models. We tried some small channel models, such as 16 channel/8 channel, and finally selected 8 channel. We also compared the difference between full-precision and half-precision training, and finally chose to use f16 to save the model parameters. The final parameters of the model we submitted were 63244 and the size was 123k. which achieved a 76.1% correct rate on the validation set.

## 5. REFERENCES

[1] Daniel S. Park , William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, Proc. Interspeech 2019, 2613-2617

[2] Terrance DeVries, Graham W. Taylor, Improved Regularization of Convolutional Neural Networks with Cutout, Computer Vision and Pattern Recognition (cs.CV)

[3] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-Excitation Networks, Computer Vision and Pattern Recognition (cs.CV)

[4] Learning Efficient Convolutional Networks through Network Slimming https://arxiv.org/abs/1708.06519

[5] 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). 2020. Submitted. URL: https://arxiv.org/abs/2005.14623.

[6] Irene Martín-Morató, Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems. 2021. arXiv:2105.13734.