# ACOUSTIC SCENE CLASSIFICATION WITH DEVICE MISMATCH USING DATA AUGMENTATION BY SPECTRUM CORRECTION

Technical Report

*Peiyao Wang, Zhiyuan Cheng, Xinkang Xu, Xinhui Hu*

Hithink RoyalFlush Information Network Co.,Ltd
No. 18, Tongshun Road, Wuchang Street, Yuhang District, Hangzhou, Zhejiang, China
wangpeiyao@myhexin.com

## ABSTRACT

This report describes the submissions by RoyalFlush of DCASE2020 task1a. Our aim is to find an audio scene classification system that is robust against multiple devices. We use log-Mel and its first and second derivatives as input features. We use the fully convolutional deep neural networks as classification model, and some strategies such as pre-Act, L2 regularization, dropout and feature normalization were applied. For improving the data imbalance caused by the different device, we tried to generate more training data by using device-related spectrum correction method.

***Index Terms***— Acoustic Scene Classification, spectrum correction, Convolutional Neural Network (CNN), DCASE

## 1. INTRODUCTION

Acoustic scene classification (ASC) is defined as recognition of the environment based on the acoustic scene which is assumed to be a valid characterization of a location or situation. Furthermore, it is assumed to be distinguishable from other scenes based on its acoustic properties [1]. This technology also has great potential for commercial application. Amazon and Google have already applied such technologies in their voice AI systems. These algorithms for these applications are embedded in commercial smart devices with microphones to recognize acoustic contextual information.
The basic framework of ASC system includes feature extraction and classification. These two steps are the key to the effectiveness of the algorithm. The most common features of ASC is log-mel spectrogram. In addition, Constant Q Transform (CQT) spectrogram, Mel-Frequency Cepstrum Coefficients (MFCCs) are also often used. For classification, early methods mostly used traditional classifiers such as Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Support Vector Machine (SVM) and Neural Network. These are the principal methods of DCASE2013. But more and more participants applied deep learning (DNN) methods such as Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) since DCASE2016. At present, deep learning is more effective and has become the principal method to realize the classification tasks. Additionally, Convolutional-Recurent Neural Network (CRNN), which has become a standard model applied for acoustic event detection, was also proved effective in recent years of DCASE [2].
The DCASE2020 task1a [3] aims at audio scene classification from multiple devices (real and simulated). It is required to classify a test audio recording into one of ten known acoustic scene classes. This task aims at generalizing system's properties across different devices, and conducts classification for the audio data produced by these devices.
Although CNN have become the most popular architecture for audio detections and other related machine learning tasks, its classification effect will be seriously degraded when the distribution mismatch exists between the training and test data. DCASE2020 task1 is concerned to deal with this problem, several different devices, including real and simulated device, were used for recording. Due to different spectral responses of devices, the data distribution among different devices of the same class may vary greatly, and which is great challenge for being classified.

## 2. DATA PROCESSING

### 2.1. Audio processing

Each audio length in the development set is 10 seconds with sampling rate of 44.1kHz. We use the log-mel spectrogram as the input feature, FFT length is 2048 with a 25% window overlapping. Finally, we get a mel-spectrogram with size 288×128(We also tried 256 dimensional mel-spectrum and 0.75 overlap, but the result did not improve significantly). Meanwhile, log-mel deltas and delta-deltas have been applied too. Our implementation was realized by using python, and the LibROSA library

### 2.2. Data augmentation

Though most ASC systems can accurately classify training samples, they suffer from inferring test records [4]. DCASE2020 task1a added 11 simulated devices in addition to 4 real devices and the sample quantity of other device is much smaller than that of device A in training set. The mismatch of sample distribution and sample imbalance between devices make this classification task difficult. So, in order to improve generalization, additional samples are generated and added into the database.
We were inspired by the work of championship team of DCASE2019 task1b [5][7], that spectrum correction has been used for data augmentation. Taking the device A as the reference device, the spectrum correction coefficients of the other devices

were calculated. Then, the data of device A and target device correction coefficients were used to simulate the data for the corresponding device.

We selected 300 audio data which were recorded at the same time and place, but by different devices. Then, we took the short-time Fourier transform on these data and obtained two sets of spectrogram with size 300*288*2048. Correction coefficients are computed by calculating the spectrogram ratio of target device and reference device. And average it in the time domain, and average it across samples, we will get a spectrum correction coefficients with size 2048.

When we get the spectrum correction coefficients of each devices, the data of device A and Spectrum correction coefficients are used to generate more simulation data of other device for training. We used this method to generate an additional 30% of the samples and expect it to improve the model's ability to recognize samples collected from the other devices.

In our experiments, we also explored the mix-up augmentation with α = 0:2, which comes from research on image classification [6]. All the input frames are normalized using the training set mean and standard deviation.

## 3. METHODS

### 3.1. Architectures

Our task is somewhat similar to the image recognition, but not entirely consistent. These tasks (including acoustic scene classification and image recognition) are represented in different physical spaces. Image is in a two-dimensional spatial domain, while the spectrogram of acoustic scenes is expressed in the time-frequency domain. The common methods for these tasks are CNN or CRNN. Through experimental comparisons, we finally adopted the whole CNN architecture as the classifier. Its overall architecture basically followed the VGG model, with a more simplified form. Two models, a large model and a small model, were used in our work. There was no big differences in performance between them. However, the small model was found more stable than the large one.

ReLU activation and BatchNormalization were used in the convolutional layers. Meanwhile, pre-activation and L2 regularization have been applied and L2 regularization's weight was set to 10-4. The specifications of these two network architectures are shown in Table1 and Table2.

### 3.2. Training

Models were trained using Adam with batch size of 32, and the cross-entropy loss function until convergence. The learning rate was reduced by 0.9 per 2000 steps. At the beginning, the data of the development set was divided into training set and verification set at a ratio of 9:1. Then, after the model is selected, the model is finally retrained for submission using all the development set data.

### 3.3. Ensemble

Ensemble is the combination of several weak models in order to get a stronger model. The idea is that even if one weak classifier gets a wrong prediction, other weak classifiers can correct the error. The generalization capacity of ensemble models is stronger than single model and it can improve the stability of results, which is why ensemble is widely studied and applied. We used two strategies: averaging and weighing voting. Experiments have shown that Ensembles of CNNs improves system performance significantly.

Table 1: Architecture of the small network

| **Layer** | **Settings** |
|---|---|
| *Input* | 288×128×3 |
| *Conv* | 16×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *Conv* | 32×Conv2D+BN+Relu (kernel 3×3, strides 2×2) |
| *Conv* | 32×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 2×2) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *GlobaPooling* | GlobaAveragePooling |
| *Softmax* | Dense(10, softmax) |

Table 2: Architecture of the large network

| **Layer** | **Settings** |
|---|---|
| *Input* | 288×128×3 |
| *Conv* | 16×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 16×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *Conv* | 32×Conv2D+BN+Relu (kernel 3×3, strides 2×2) |
| *Conv* | 32×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 2×2) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 64×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Pooling* | MaxPooling(2×2) |
| *Conv* | 128×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 128×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *Conv* | 10×Conv2D+BN+Relu (kernel 3×3, strides 1×1) |
| *GlobaPooling* | GlobaAveragePooling |
| *Softmax* | Dense(10, softmax) |

## 4. RESULT

DCASE officially divides the training set and the test set when releasing the development set. Here are the results of our system based on this dataset. Table 3 presents the class-wise results and Table4 presents the device-wise results over testing set, which is a subset of the development set on the task 1a, and it shows that the accuracy is improved compared to the DCASE2020 baseline.

Table3: Accuracy on the fold 1 evaluation set (class-wise)

| **Scene label** | **Baseline** | **Our system** |
|---|---|---|
| Airport | 45.0% | 56.6% |
| Bus | 62.9% | 71.2% |
| Metro | 53.5% | 65.6% |
| Metro station | 53.0% | 60.1% |
| Park | 71.3% | 79.6% |
| Public square | 44.9% | 55.7% |
| Shopping mall | 48.3% | 56.9% |
| Street pedestrian | 29.8% | 51.2% |
| Street traffic | 79.9% | 78.4% |
| tram | 52.2% | 59.4% |
| **Average** | **54.1%** | **63.9%** |

Table4: Accuracy on the fold 1 evaluation set (device-wise)

| **Device** | **Baseline** | **Accuracy** |
|---|---|---|
| A | 70.6% | 74.5% |
| B | 60.6% | 63.3% |
| C | 62.6% | 70.3% |
| S1 | 55.0% | 65.5% |
| S2 | 53.3% | 62.7% |
| S3 | 51.7% | 63.9% |
| S4 | 48.2% | 60.6% |
| S5 | 45.2% | 63.9% |
| S6 | 39.6% | 50.6% |

## 5. SUBMISSIONS

We use the average of different combinations of the models mentioned above trained with different strategies for our final submissions. The test accuracy of these combinations on development set are shown in Table 5

Table5: Accuracy of fusion system on the fold 1 evaluation set

| System name | Subsystem count | Accuracy |
|---|---|---|
| Wang_RoyalFlush_task1a_1 | 6 | 63.9% |
| Wang_RoyalFlush_task1a_2 | 3 | 62.9% |
| Wang_RoyalFlush_task1a_3 | 4 | 62.1% |
| Wang_RoyalFlush_task1a_4 | 6 | 62.7% |

## 6. CONCLUSION

In this technical report, we detailed our approaches to accomplish the DCASE2020 Task1a challenge. We used the full CNN model and did a lot of optimization experiments in terms of training strategies and parameters. We use spectrum correction method to augment more device-related data and mitigate the impact of mismatched data distribution caused by multiple devices. The results has improved by 9.8% over the baseline.

## 7. REFERENCES

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalder, A. Shah, E. Vincent, B. Raj, T. Virtanen, " DCASE 2017 challenge setup: Tasks, datasets and baseline system," DCASE2017 Challenge, Tech. Rep., Nov. 2017

[2] Pham L, Doan T, Ngo D, et al "CDNN-CRNN Joined Model for Acoustic Scene Classification" Tech. Rep., 2019, DCASE 2019 technical reports.

[3] http://dcase.community/challenge2020/

[4] Chen H, Liu Z, Liu Z, et al. "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling" Tech. Rep., 2019, DCASE 2019 technical reports.

[5] Kosmider M. Calibrating neural networks for secondary recording devices[C]//Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), New York, NY, USA. 2019: 25-26.

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," CoRR, vol. abs/1710.09412,2017.[Online].Available:http://arxiv.org/abs/1710.09412

[7] Nguyen T, Pernkopf F, Kosmider M. Acoustic Scene Classification for Mismatched Recording Devices Using Heated-Up Softmax and Spectrum Correction[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 126-130.