# ACOUSTIC SCENE CLASSIFICATION WITH VARIOUS DEEP CLASSIFIERS

Technical Report

*Yue Liu, Xinyuan Zhou, Yanhua Long*

Shanghai Normal University, Shanghai, China
lliuyuely@163.com, xinyuan_zhou@yeah.net, yanhua@shnu.edu.cn

## ABSTRACT

In this report, we describes the SHNU team's submission to the DCASE-2020 challenge Task1-A (Acoustic Scene Classification with Multiple Devices). In our submissions, three different deep models are investigated. The first one is a ResNet-based model with receptive-field regularization. The second one is a common two-dimensional CNN model with perceptual weighted power spectrogram as input. The third one is a self-attention based model with only Transformer encoder architecture which is specially designed for acoustic scene classification. In addition, we proposed a device-enhancement data augmentation method, together with the conventional mix-up and specAugment to improve the model robustness to multiple devices. Experimental results on the fold1 validation set show that these models are complementary in some extent. We prepared all of our submissions *without the use of any external data except for the official baseline embeddings*. The logistic regression score fusion is used to fuse the softmax outputs of single-systems.

***Index Terms***— Acoustic Scene Classification, ResNet, CNN, Transformer

## 1. INTRODUCTION

This technical report describes our submissions to the Task1 (Subtask A, Acoustic Scene Classification with Multiple Devices) in the DCASE-2020 challenge. First,we retrained the official baseline as one of our single-systems for score fusion. It has two fully-connected feed-forward neural network layers with OpenL3 embeddings as its inputs. Then, we train a ResNet model that has been proposed in [1] as our first submitted single-system. To improve the model generalization ability, we design a device-enhancement data augmentation method. This method together with the conventional mix-up [2] are used to enhance the ResNet model.

Our second submitted single-system is a 9 layers convolutional neural networks (CNN-9) with $2 \times 2$ average pooling, using the perceptual weighted power spectrogram of each audio recording as input. Our implementation is based on the CNN-9 code scripts [1] provided by Qiuqiang Kong, et.al [3]. For this system, only mix-up is used for data augmentation.

Our third submitted single-system is a self-attention based model (E-Transformer), in which two LSTM layers are used to model the high-level representations from the Transformer encoder. More detail of these single-systems will be described in next sections. Based on these four single-systems, our final four submissions for the Task1-A are: (1) ResNet; (2) CNN-9; (3) E-Transformer, (4) Softmax score fusion of the official baseline, ResNet and CNN-9. Experimental results on the fold1 validation set show that the system (1)-(4) achieve the classification accuracy of 70.24%, 68.82%, 58.15%, 73.13% respectively.

[1] https://github.com/qiuqiangkong/dcase2019_task1

## 2. DATA PREPARATION

Each audio recording is first down-sampled from the original 44.1 Khz to 22.05 Khz. Then we extract the acoustic features using a Short-time Fourier Transform (STFT) with a window size of 2048-sample and a hop-size of 512-sample. These STFTs are further perceptually weighted using a 256-bin Mel filter bank to obtain the final perceptual weighted power spectrograms.

### 2.1. Data Augmentation

We tried both the conventional mix-up [2] and specAugment [4] methods to enhance the model robustness. Moreover, to handle training and testing mismatch between multiple devices, we propose a simple device-enhancement data augmentation method (deviceAugment) to enhance the neural network model training. Figure 1 illustrates the simple framework of the proposed *deviceAugment* method.

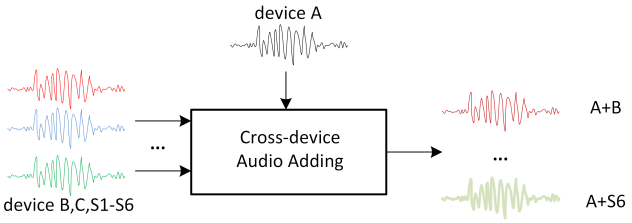


Figure 1: Framework of the proposed deviceAugment method.

As shown in Figure 1, we take the device A recording as reference, and then add each recording from device B, C, and S1-S6 to the recording from device A at 5dB signal-to-noise ratio using Kaldi toolkit [5]. These *deviceAugment* recordings are then combined with the official development training data to train each single-system.

## 3. ARCHITECTURES

### 3.1. ResNet

Our ResNet model architecture is the same CNN architecture as the RN2 described in Table 1 of work [1]. The optimal reception field used in our model is $87 \times 87$ pixels over the extracted spectrograms.

For the model optimization, we used Adam with a Adam(0.9, 0.99) scheduler. We start training with a learning rate of $1 \times 10^{-4}$. From epoch 100 until 150, the learning rate decays linearly from $1 \times 10^{-4}$ to $1 \times 10^{-5}$. The rest 200 epochs are with the learning rate $1 \times 10^{-6}$. No cross-validation is performed in our system training.

### 3.2. CNN-9

The basic architecture of our CNN-9 model is similar to the work in [3]. However, we changed the model parameters as shown in Table 1. It has four convolution blocks (*ConvBlock*), each block contains convolution layer, batch normalization layer and ReLu activation layer. After the convolution layers, the AvgPooling was performed on the Frequency dimension and the MaxPooling was performed on the time dimension. Finally, we used a fully connected layer with softmax function to get the prediction score of each acoustic class.

Table 1: Our CNN-9 architecure. BN: Batch Normalization. ReLu: Rectified Linear Unit.

| Name | Description | Output size |
|---|---|---|
| Input | Channel × Time × Frequency | $1 \times 256 \times 431$ |
| ConvBlock1 | Cov3 × 3 -64BN-ReLu<br>Cov3 × 3 -64BN-ReLu<br>AvgPooling 2 × 2 | $64 \times 128 \times 215$ |
| ConvBlock2 | Cov3 × 3 -128BN-ReLu<br>Cov3 × 3 -128BN-ReLu<br>AvgPooling 2 × 2 | $128 \times 64 \times 107$ |
| ConvBlock3 | Cov3 × 3 -256BN-ReLu<br>Cov3 × 3 -256BN-ReLu<br>AvgPooling 2 × 2 | $256 \times 32 \times 53$ |
| ConvBlock4 | Cov3 × 3 -512BN-ReLu<br>Cov3 × 3 -512BN-ReLu<br>AvgPooling 1 × 1 | $512 \times 32 \times 53$ |
| AvgPooling | AvgPooling 32 × 1 | $512 \times 53$ |
| MaxPooling | MaxPooling 53 × 1 | 512 |
| FC | Linear(512,10)<br>Softmax | 10 |

### 3.3. E-Transformer

Our E-Transformer model is based on the Transformer encoder with multi-head self-attention mechanism. We perform the same down-sampling as in [9] before the encoder, using two $3 \times 3$ CNN layers with stride 2 to reduce the GPU memory occupation and and the length of the input sequence. The detail architecture is shown in Figure 2.

The experiments was conducted on ESPnet[6] end-to-end speech processing toolkit. We extract 120-dimensional log Mel-filter bank as acoustic features and normalize them with global mean computed from the training set. The frame-length is 64 ms with a 20 ms shift. In our experiment, model contains 8-layer encoder, where the $d_{\text{model}} = 256$ and the dimensionality of position-wise Feed-Forward Networks $d_{\text{ff}} = 512$. In all attention sub-layers,16 heads are used. The whole network is trained for 400 epochs and warmup[7] is used for the first 4,000 iterations. Also the SpecAugment[8] is used for data augmentation.

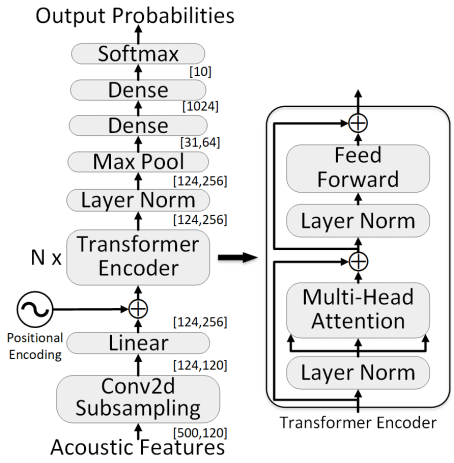


Figure 2: Framework of the proposed E-Transformer system.

## 4. RESULTS AND DISCUSSIONS

In this section, all the results are reported on the fold1 validation set, and the models are trained using the development training data as the official baseline of DCASE 2020 Task1-A.

### 4.1. Results on Single-systems

Table 2: Classification accuracy (%) on the development set of our single-systems.

| System | Data Augmentation | Acc(%) |
|---|---|---|
| Baseline | - | 53.8 |
| ResNet | - | 68.5 |
| | mix-up | 69.5 |
| | mix-up + deviceAugment | 70.2 |
| | mix-up + specAugment | 68.9 |
| CNN-9 | - | 67.3 |
| | mix-up | 68.8 |
| | mix-up + deviceAugment | 68.7 |
| | mix-up + specAugment | 67.1 |
| E-Transformer | specAugment | 58.2 |

Table 2 shows the classification accuracy of each single-system. We reproduced the official baseline system (with OpenL3 embedding) and got a 53.8% accuracy on the fold1 development set. For the ResNet and CNN-9 systems, we tried to perform the mix-up, the *deviceAugment* and the *specAugment* for training data augmentation. However, we find that the *specAugment* is not effective for both of our ResNet and CNN-9 models. The *deviceAugment* can slightly improve the ResNet. And we observed that the specAugment is very important to improve the E-Transformer model. Therefore, we choose the setup that achieved the best performance on the

fold1 development to train our final single-systems for the challenge submission.

### 4.2. System Fusion

For the system fusion, we simply fuse the single-systems at score level using the Bosaris toolkit [10], details on the fold1 development set are shown in Table 3. We find that the combination of the baseline, CNN-9 and ResNet achieves the best result.

Table 3: Classification accuracy (%) on the development set of score fusion systems. The 'Baseline', 'ResNet' and 'CNN-9' represent the official baseline with OpenL3 embedding, the RestNet with *mixup* and *deviceAugment* data augmentation, and the CNN-9 with *mixup* respectively.

| System | Acc(%) |
|---|---|
| Baseline+ResNet | 71.0 |
| Baseline+CNN-9 | 69.6 |
| Baseline+CNN-9+ResNet | 73.1 |

## 5. CONCLUSION

In this report, we detailed our approaches to tackle the DCASE2020 Task1-A challenge. We showed that our ResNet and CNN-9 models significantly outperformed the official baseline. Unfortunately, the proposed attention and Transformer encoder dependent system (E-Transformer) only achieved small improvement over the baseline. The detail setups of each submission single-systems and the behavior of system fusion at the score level are also described.

## 6. ACKNOWLEDGMENT

Thanks to the National Natural Science Foundation of China (No.61701306) for funding.

## 7. REFERENCES

[1] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification," in *Proc. EUSIPCO*,2019, pp. 1–5.

[2] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopezpaz, "Mixup: Beyond Empirical Risk Minimization," in *Online, available: http://arxiv.org/abs/1710.09412*.

[3] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, M, Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," in *arXiv preprint arXiv:1904.03476 (2019)*.

[4] D.S. Park, W. Chan, Y. Zhang, et.al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *CoRR abs/1904.08779*, 2019.

[5] D. Povey, A. Ghoshal, G. Boulianne and et.al., "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[6] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Proc. INTERSPEECH*, 2018, pp. 2207–2211.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[8] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple augmentation method for automatic speech recognition," in *Proc. INTERSPEECH*, 2019, pp. 2613–2617.

[9] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based end-to-End speech recognition with connectionist temporal classification and language model integration," in *Proc. INTERSPEECH*, 2019, pp. 1408–1412.

[10] N. Brümmer, E. Villiers. "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF", in *arXiv:1304.2865v1*, 2013.