Designing Acoustic Scene Classification Models with CNN Variants

Technical Report

Sangwon Suh, Sooyoung Park, Youngho Jeong, Taejin Lee[*]

Media Coding Research Section
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea
{suhsw1210, sooyoung, yhcheong, tjlee}@etri.re.kr

ABSTRACT

This technical report describes our Acoustic Scene Classification systems for DCASE2020 challenge Task1. For subtask A, we designed a single model implemented with three parallel ResNets, which is named Trident ResNet. We have confirmed that this structure is beneficial when analyzing samples collected from minority or unseen devices, and confirmed 73.7% classification accuracy for the test split. For subtask B, we used the Inception module to build a model named Shallow Inception that has fewer parameters than the CNN of the DCASE baseline system. Due to the sparse structure of the Inception module, we have enhanced the accuracy of the model up to 97.6%, while reducing the number of parameters.

Index Terms— Acoustic Scene Classification, Convolution Neural Network, Residual block, Inception module, ResNet

1. INTRODUCTION

Acoustic Scene Classification (ASC) is a task of classifying given data to a place where it was recorded. Each data corresponds to one class out of ten, and there is no data with multiple labels. The length of the data is ten seconds, but the useful information appears very rarely. This task is one of the major topics that has been covered every year in the DCASE challenge. This year, the ASC task was released in two subtasks: Subtask A for multiple devices dataset, and Subtask B for designing the Low-complexity model [1].

The main issue of the subtask A is to design a classifier that works stably on various microphone types. However, the development dataset mostly includes the data collected from a specific microphone, and the evaluation data will include data recorded with a microphone that has not appeared in the development set. This task was treated in the previous year, and [2] was placed on top with spectrum correction method and Convolutional Neural Network (CNN) model.

In the case of subtask B, the audio files in the dataset are identical to the previous year, but only change in labels: ten classes were merged into three. The main issue of this task is to design a model under 500 kilobytes. This corresponds to 128K when converted to a 32-bit floating-point per parameter. It is very small number considering last year's participants submitted more than millions or billions of parameters.

The following sections include details of our model structure and training methods. Due to the model size limitation in subtask B, it became impossible to solve both problems with a universal model design. Therefore, we separated the descriptions for subtask A and B if necessary.

2. DATASETS

2.1. Subtask A: TAU Urban Acoustic Scene 2020 Mobile

The development dataset of TAU Urban Acoustic Scene 2020 Mobile, which contains 23,040 samples, was used to train and validate the model. This dataset consists of various audio samples collected from three real devices and six simulated devices. Most of the data were collected from device A, a binaural microphone, and data from Samsung Galaxy S7 and iPhone SE are also included. According to the organizer's report, the evaluation dataset will include samples from GoPro Hero5 Session. The simulated devices are synthesized by processing the data of device A with various impulse responses and dynamic range compression. The organizer of the challenge provides basic metadata of training/test split consisting of 13,965 samples in the training set and 2,970 samples in the test set.

2.2. Subtask B: TAU Urban Acoustic Scenes 2020 3Class

The dataset of subtask B is the development dataset of TAU Urban Acoustic Scenes 2020 3Class, which contains 14,400 samples. This is the same dataset used in Subtask A of DCASE2019 that consists of ten different acoustic scenes from twelve European cities. The only change is that ten labels have been changed into three labels: indoor, outdoor, and transportation. The organizer of the challenge provides basic metadata of the training/test split consisting of 9,185 samples in the training set and 4,185 samples in the test set.

* Thanks to Korea government (MSIT) for funding.

3. SYSTEM ARCHITECTURE

3.1. Data Preprocessing

3.1.1. Subtask A: log Mel spectrogram with deltas/delta-deltas

The data of subtask A are mono audio files with 44.1 kHz sample rate. We transformed them into power spectrogram by skipping every 1024 samples with 2048 length Hann window. A spectrum of 431 frames was yielded from 10 seconds audio file, and each spectrum was compressed into 256 bins of Mel frequency scale. Additionally, deltas and delta-deltas were calculated from the log Mel spectrogram and stacked into the channel axis. The number of frames of the input feature is cropped by the length of the delta-delta channel so that the final shape becomes $[256 \times 423 \times 3]$.

3.1.2. Subtask B: log Mel spectrogram

The data of subtask B are stereo audio files with 48 kHz sample rate. We transformed them into log Mel spectrogram with the same strategy we've conducted on subtask A, without deltas and delta-deltas. The final shape of input feature data is $[256 \times 469 \times 2]$.

3.2. Data Augmentations

We only utilized training split of the challenge dataset, and applied data augmentation to increase the diversity of data distribution. Our data augmentation strategies are listed in Table 1. The augmented data were generated from each mini-batch consisting of 64 samples during the training process in real-time.

Table 1: List of data augmentation strategies

Strategy	Parameter
Temporal crop	Crop length = 5 seconds
Mixup [3]	Alpha = 2.0

3.3. Model Design

3.3.1. Subtask A: Trident ResNet model

Previous studies have verified the effectiveness of the ResNet [4] on the ASC [5, 6, 7]. A Residual block in our model consists of two 3×3 convolution blocks sequentially and an identity path with zero-padding after average pooling as shown in Figure 1. Each convolution block is pre-activation convolution so that the layer order is BatchNormalization-ReLU-Convolution. Gamma and beta terms are not used in Batch Normalization layers except for the first layer, and there is no bias term in Convolution layers. Kernels are initialized with He normal distribution [8] and regularized with L2 regularization of 5×10^{-4}. Detailed descriptions are written in the following subsections.

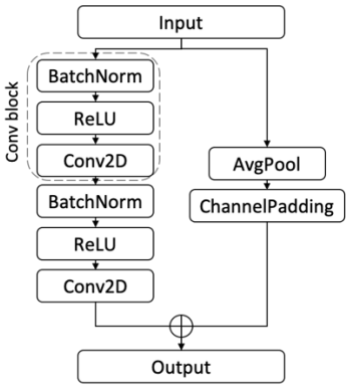

Figure 1: Residual block with pre-activation

The previous study [9] claimed that the proper size of the receptive field is crucial for the ASC task. They confirmed that the CNN with a large receptive field is overfitting for ASC data, and proposed a method to increase the classification performance by restricting the receptive field. Also, they evaluated the frequency and frame axis separately, and found that the model performance is sensitive to the size of the frequency axis. Similar concept can be seen in the model structure of [6]. Their model reduces the time information by striding of the convolution filters but preserves the frequency bins. Inspired by the model structure of [6], we conducted a grid search to find the appropriate receptive field for the input feature prepared in 3.1.1. We adjusted the receptive field size of our ResNet by stacking the residual blocks: the deeper the network, the wider the receptive field. Additionally, we introduced the frequency-wise dilated convolution layers after the frame-wise stride convolution layers to increase the receptive field size on the frequency axis. The detailed configurations of our ResNet are shown in Table 2.

Table 2: Block configurations of our ResNet.

Block name	Configuration
Input	
BatchNorm	Learn γ and β
Conv2D	(1×2) strides
Residual	
Residual	(2×1) dilation on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(1×2) strides on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(1×2) strides on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(1×2) strides on the first Conv
Residual	(2×1) dilation on the first Conv
Residual	(2×1) dilation on the first Conv
Output	

We arranged the ResNets in parallel and concatenated their outputs for classification. This parallel structure has been proposed in [6] and [10] to learn distinct features from different frequency bands. Our model consists of three paths for 0-63, 64-127, and 128-255 Mel bins. We've also evaluated the dual parallel structure, which splits the Mel bins in half, but the triple architecture

performed better for minority/unseen devices. After concatenating the outputs from each network, two blocks of 1×1 convolution and Global Average Pooling (GAP) calculates the classification scores. The overall structure of our model is shown in Figure 2.

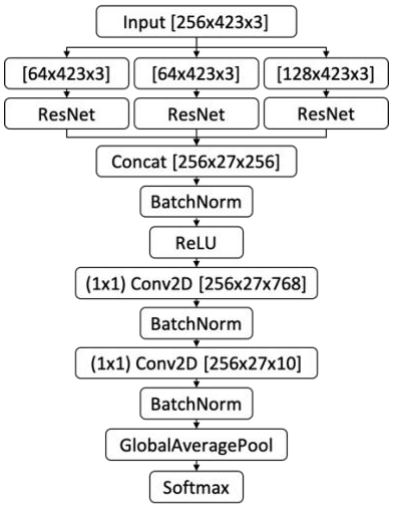

Figure 2: Overall structure of *Trident ResNet* model.

3.3.2. Subtask B: Shallow Inception model

The Inception module is a convolution block that prevents the overfit by reducing the parameters with sparse connectivity [11]. We've confirmed that the Inception module proposed in [12] performed better than the original structure of GoogLeNet. The proposed module composes a pooling path with average pooling as shown in Figure 3.

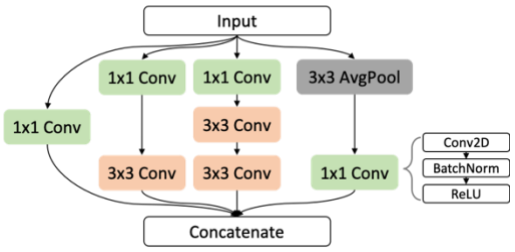

Figure 3: Inception module with dimension reduction

We implemented Shallow Inception model by stacking blocks to the depth where the performance is saturated within limited parameters. And to compensate for the flexibility of the model due to insufficient depth, the Batch Normalization layer was configured to learn the beta terms. Kernels are initialized with He normal distribution and regularized with L2 regularization of 1×10^{-4}. Table 3 is the overall structure of our model.

Table 3: Overall structure of *Shallow Inception* model

Block name	Configuration	Output shape
Input		$[256 \times 469 \times 2]$
BatchNorm	Learn γ and β	$[256 \times 469 \times 2]$
Conv2D	(1×2) strides	$[256 \times 235 \times 64]$
BN-ReLU	Learn β	$[256 \times 235 \times 64]$
Inception		$[256 \times 235 \times 128]$
AvgPool	(1×3) pooling	$[256 \times 78 \times 128]$
Inception		$[256 \times 78 \times 160]$
Conv2D	(1×1) kernel	$[256 \times 78 \times 128]$
BN-ReLU	Learn β	$[256 \times 78 \times 128]$
Conv2D	(1×1) kernel	$[256 \times 78 \times 3]$
BatchNorm	Learn γ and β	$[256 \times 78 \times 3]$
GAP		[3]
Output	Softmax	[3]

3.4. Categorical Focal Loss

Focal loss [13] attenuates the log-loss generated by well-trained samples, so that the model can focus on the poorly trained samples. The following equation describes focal loss with balancing parameter α, focusing parameter γ and prediction score p_t,

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t) \quad (3)$$

Increasing the value of γ increases the sensitivity of the model to misclassified samples, and α scales the loss function linearly. Our setting was 2.0 and 0.25, respectively.

3.5. Training Setup

We trained our model using Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9. The learning rate is controlled by a cosine annealing schedule and restarts with initial learning rate *lr* at 2, 6, 14, 30, 126, and 254 epochs. The value of *lr*, which is 0.1 at 0 epoch, decreases by 10% for each restart. The cosine scheduler decays the learning rate to $lr \times 10^{-4}$, so the model can explore deeper areas on the hyperplane for each restart.

3.6. Snapshot ensemble [14]

We saved snapshots every cycle of the training process, and combined them to build an ensemble model that outperforms a single model. The submitted ensemble systems of subtask A consisted of models trained at 62, 126, and 254 epochs. And the submitted ensemble systems of subtask B consisted of models trained at 254 and 510 epochs. The scores from each model were averaged, or weighted averaged to make ensemble prediction.

4. RESULTS

This section reports the average of the class-wise accuracies of our submitted systems for the train/test split. System 1 of each subtask was trained with the provided train split, while the other systems 2, 3, and 4 were trained with the entire development set. Therefore, the results for the test split of system 1 and 2 are the same.

Table 4: Test split results of subtask A development set

ID	System name	Accuracy
-	DCASE2020 Task1 Baseline, Subtask A	54.1 %
1	TridentResNet_DevSet	73.7 %
2	TridentResNet_EvalSet	73.7 %
3	TridentResNet_Ensemble	74.2 %
4	TridentResNet_Weighted_Ensemble	74.4 %

Table 5: Test split results of subtask B development set

ID	System name	Accuracy
-	DCASE2020 Task1 Baseline, Subtask B	87.3 %
1	ShallowInception_DevSet	97.6 %
2	ShallowInception_EvalSet	97.6 %
3	ShallowInception_Ensemble	97.5 %
4	ShallowInception_Weighted_Ensemble	97.7 %

5. ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00050, Development of Human Enhancement Technology for auditory and muscle support).

6. REFERENCES

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

[2] Kosmider, M. (2019, June), "Calibrating neural networks for secondary recording devices," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, New York, NY, USA (pp. 25-26).

[3] H. Zhang, M. Cisse, Y. N. Dauphin, and D. LopezPaz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[4] He, K., Zhang, X., Ren, S., and Sun, J. (2016), "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[5] Koutini, K., Eghbal-zadeh, H., and Widmer, G. (2019), "CP-JKU submissions to DCASE'19: Acoustic Scene Classification and Audio Tagging with Receptive-Field-Regularized CNNs," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*.

[6] Gao, W., and McDonnell, M. (2019), "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," *DCASE2019 Challenge, Tech. Rep.*

[7] Liu, M., Wang, W., and Li, Y, "THE SYSTEM FOR ACOUSTIC SCENE CLASSIFICATION USING RESNET," *DCASE2019 Challenge, Tech. Rep.*

[8] He, K., Zhang, X., Ren, S., and Sun, J. (2015), "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

[9] Koutini, K., Eghbal-Zadeh, H., Dorfer, M., and Widmer, G. (2019, September), "The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification," in *2019 27th European Signal Processing Conference (EUSIPCO)* (pp. 1-5), IEEE.

[10] Phaye, S. S. R., Benetos, E., and Wang, Y. (2019, May), "SubSpectralNet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 825-829), IEEE.

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... and Rabinovich, A. (2015), "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[12] Suh, S., Lim, W., Park, S., & Jeong, Y, "Acoustic Scene Classification Using SpecAugment and Convolutional Neural Network with Inception Modules," *DCASE2019 Challenge, Tech. Rep.*

[13] Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017), "Focal loss for dense object detection," In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

[14] Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017), "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109.*