Acoustic Scene Classification Using SpecAugment and Convolutional Neural Network with Inception Modules

Technical Report

Sangwon Suh, Wootaek Lim, Sooyoung Park, Youngho Jeong

Realistic AV Research Group
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea
suhsw1210@etri.re.kr

ABSTRACT

This paper describes the system submitted to the Task 1a (Acoustic Scene Classification, ASC). By analyzing the major systems submitted in 2017 and 2018, we have selected a two-dimensional convolutional neural network (CNN) as the most suitable model for this task. The proposed model is composed of four convolution blocks; two of them are conventional CNN structures but the following two blocks consist of Inception modules. We have constructed a meta-learning problem with this model in order to train the super learner. For each base model training, we have applied different validation split methods to take advantage in generalized result with the ensemble method. In addition, we have applied data augmentation in real time with SpecAugment, which was performed for each base model. With our final system with all of the above techniques have applied, we have achieved an accuracy of 76.1% with the development dataset and 81.3% with the leader board set.

Index Terms— DCASE 2019, Acoustic scene classification, Convolutional neural network, Inception module, SpecAugment, Ensemble method, Super learner, Meta-learning

1. INTRODUCTION

Acoustic scene classification is a task of analyzing audio recordings and classifying them into ten predefined classes. In DCASE2019 challenge [1], it was announced as Task 1 that includes three subtasks. This report is targeted at subtask A, which is a classification of ten acoustic scenes acquired in twelve European cities with the same recording device.

On the study of Sakashita et al., which scored the highest classification accuracy in last year's DCASE challenge, have resolved this task with the convolutional neural network in the spectrogram domain [2]. This is an application of conventional image classification models to the audio signal by converting it into an image domain. This approach can be seen in other studies like [3] and [4].

In this report, we have proposed a CNN model consist of the Inception modules [5, 6, 7] and evaluated the classifiers which are trained with various training settings. Moreover, we have introduced an ensemble method by combining classifiers to enhance performance. The following section will cover a detailed explanation of system architecture, experiments, experimental results, and conclusion.

2. SYSTEM ARCHITECTURE

2.1. Feature extraction and normalization

Our system utilizes a log-amplitude mel-spectrogram as model input. First, each audio file is loaded with a sample rate of 48 kHz and merged into mono. After then, the audio data is converted into spectrogram with short-time Fourier transformation, which is computed on 40 milliseconds Hamming window with 20 milliseconds overlap using 2048-point FFT. The spectrogram is converted to 128 bands mel scaled features and transformed into log-scale. The size of the input feature finally obtained is 500 frames in time, and 128 bands in frequency. All features are mean-centered and normalized along with the individual frequency bins.

2.2. Data augmentation

TAU Urban Acoustic Scenes 2019 development dataset contains only data comes from 10 of the 12 cities. Therefore, overfitting on the development dataset will result in poor classification accuracy for the two unseen cities that appear only in the evaluation dataset. We have tested the provided training/test split of the development dataset and found the classification accuracy was lower in unseen cities. To overcome this overfitting problem, we have used a method called SpecAugment to mix some noise into training data [8]. SpecAugment is a data augmentation technique that can be applied in the spectrogram domain and performs augmentation by applying three methods: time masking, frequency masking, and time stretching. Except for the time stretching method, we have used two time masks and two frequency masks per data to generate additional training data. However, this seemed to add too much noise to the training data, because the experiment resulted in poor classification accuracy. Therefore, we have changed to apply the SpecAugment only to the data that are randomly selected for each model learning stage. In this case, a class imbalance may occur between the training sets generated each time, but it can be overcome with the ensemble method.

2.3. Base model structure

The base model that we have used on this task is depicted in Figure 1. It is composed of four convolution blocks with the Inception modules. The core function of the Inception module is to construct a feature map composed of various size of receptive fields in one block so that the network can learn the optimal local sparse structure. The first and second convolution blocks correspond to the stem layers; each consists of 3x3 cascading convolution layers. And the following third and fourth convolution blocks use the Inception modules as shown in Figure 1. We tried a deep network like Inception v4, but could not improve the classification accuracy of this task. Therefore, to increase the filter map depth without additional convolution blocks, the number of filters is doubled and the 2x2 pooling layer applied to every convolution block output. We have made three models according to the pooling layer type: max, average, and mix pooling. Mix pooling is a technique that applies different pooling methods for the time axis and the frequency axis. It consists of a 2x1 max pooling followed by a 1x2 average pooling. In the feature maps generated after the four convolution blocks, it is considered that there would be some time slots that strongly represent the current class. Therefore, we have obtained the softmax value for 10 classes with time distributed dense layer after average remaining frequency information. The softmax value of each class is averaged over the time axis and used as the model output.

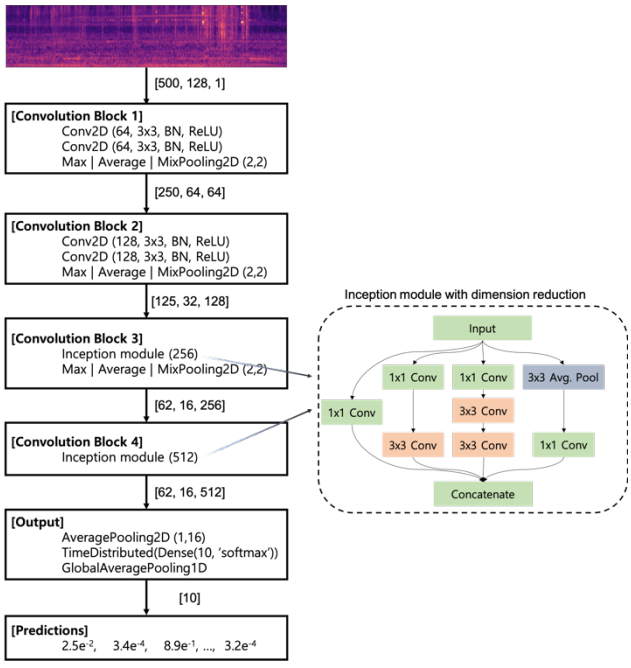

Figure 1. Base model structure

3. EXPERIMENTS

3.1. Dataset

TAU Urban Acoustic Scene 2019 development dataset was used for this experiment. It is composed of ten acoustic scenes recorded in ten large European cities and contains class-balanced 14,400 audio files. Each audio file is a 10-second stereo file with a sample rate of 48 kHz. The task organizer provides a training/test split metadata, which contains 9,185 training files and 4,185 test files. This split configures Milan to be an unseen city that only included in the test split.

3.2. Training settings

We have used Adam [9] as an optimizer with an initial learning rate of 10-3 and an epsilon value of 10-8. Each model was trained for 70 epochs with 32 samples in a mini-batch. Without making additional folds, the provided 9,185 of training data was split into various training and validation splits shown in Table 1. Balanced split type is a method of splitting validation data under certain conditions, which is controlled by balancing mode. Random split type splits without any condition and may cause location or class imbalance on validation split. Model performance is evaluated after each epoch on the validation set, and a model weight is selected at the best epoch. If there is no improvement in performance compared to the previous epoch, the learning rate is reduced by 10%. We examined the Glorot uniform [10] and He normal [11], but there was no significant difference between the two weight initializers.

Table 1. Validation split types and abbreviated indications

Split type -balanc-ing_mode	Balanced -Identifier two level hierarchy	Balanced -Identifier	Balanced -Class	Random
Indication	B-I2	B-I	B-C	R

3.3. Network ensemble method

We were able to obtain various classifiers by combining the aforementioned materials: three models, four validation split methods, and two weight initializers. It was confirmed that the performance of a particular class was different for each classifier. Therefore, we have constructed an ensemble classifier by combining the classifiers with a trainable model as shown in Figure 2. This ensemble method is Stacking, and the final decision-making model is called super learner. The super learner is trained by the predictions of weak learners and this process is called meta-learning. In our case, the softmax predictions of each classifier become the training data for the super learner. Our super learner is a weighted average model consists of 1D convolution and softmax activation layer.

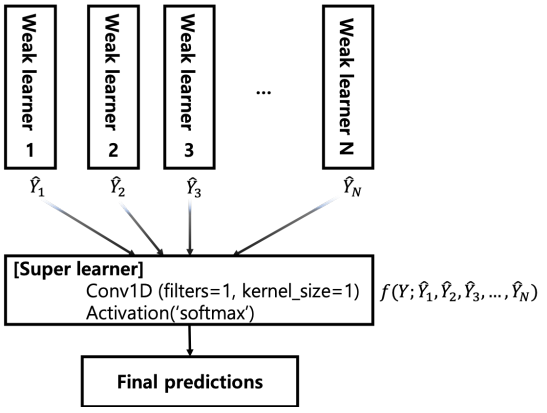

Figure 2. Ensemble classifier

4. EXPERIMENTAL RESULTS

Table 2, Table 3 and Table 4 show the results of the classifiers tested on the development dataset. There seems to be no performance variation due to the pooling method and initializer. In the case of the validation split method, Table 2 shows that the B-C is the best and the B-I2 is the worst. This means that training in various locations can guarantee performance on the unseen data. In Table 4, (9) to (16) classifiers were trained with different random seeds. For all classifiers evaluated, it was not able to find any patterns in the classification accuracy of unseen cities, so we concluded that the randomly generated data by SpecAugment contributed to it.

Table 2. Classifiers evaluation results on the development dataset with He normal initializer and 30% validation split (MA: Mix, M: Max)

Classifier	Seen (%)	Unseen (%)	Total (%)
He_MA_B-I2_30	71.8	62.5	71.1
He_MA_B-I_30	73.3	64.9	72.7
He_MA_B-C_30	73.8	66.2	73.1
He_MA_R_30	74.8	63.4	73.8
He_M_B-I2_30	73.2	51.5	71.4
He_M_B-I_30	73.1	64.8	72.5
He_M_B-C_30	74.9	56.6	73.2
He_M_R_30	73.9	63.8	73.1

Table 3. Classifiers evaluation results on the development dataset with He normal initializer and 10% validation split (MA: Mix, M: Max, A: Average)

Classifier	Seen (%)	Unseen (%)	Total (%)
(1) He_MA_B-I2_10	73.6	50.7	71.7
(2) He_MA_B-I_10	73.3	63.7	72.5
(3) He_MA_B-C_10	74.9	71.3	74.7
(4) He_MA_R_10	73.3	59.9	72.2
(5) He_M_B-I2_10	73.2	54.0	71.4
(6) He_M_B-I_10	71.1	61.4	70.2
(7) He_M_B-C_10	74.4	59.7	73.1
(8) He_M_R_10	74.0	61.7	73.1
He_A_B-I2_10	72.9	54.3	71.3
He_A_B-I_10	73.6	58.6	72.3
He_A_B-C_10	74.9	63.2	74.0
He_A_R_10	74.6	63.2	73.7

Table 4. Classifiers evaluation results on the development dataset with Glorot uniform initializer and 10% validation split (MA: Mix, M: Max)

Classifier	Seen (%)	Unseen (%)	Total (%)
GL_MA_B-I2_10	72.6	62.5	71.8
GL_MA_B-I_10	73.5	59.1	72.4
GL_MA_B-C_10	75.5	66.1	74.8
GL_MA_R_10	75.7	60.3	74.4
(9~12) GL_MA_R_10	74.1~75.6	59.1~67.9	72.9~74.8
(13~16) GL_M_R_10	73.7	56.9~64.2	73~74.2

The submitted classifiers to the DCASE2019 task1a are listed in Table 5. BASE_MODEL is a classifier obtained by re-training the base model several times and selecting the best performance on leaderboard dataset. Ensemble8 is a combination of (1) to (8) classifiers in Table 3, and Ensemble17 is a combination of BASE_MODEL, Enasemble8 and (9) to (16) classifiers in Table 4. Finally, Eensemble25 is an ensemble classifier that combines all the classifiers in Table3, Table 4, and BASE_MODEL.

Table 5. Submitted classifiers evaluation results on the development dataset and the leaderboard dataset

Classifier	Seen (%)	Unseen (%)	Total (%)	Leaderboard (%)
BASE_MODEL	75.4	62.3	74.4	79.6
Ensemble8	76.5	64.8	75.5	80.6
Ensemble17	77.0	65.9	76.1	81.1
Ensemble25	76.9	66.1	76.1	81.3

5. CONCLUSION

This paper reports our acoustic scene classification system for DCASE2019 task1a. This task evaluates the classification accuracy for ten acoustic scenes acquired in twelve European cities. However, since the development dataset only contains the audio from ten cities, it is important to guarantee the performance for unseen cities. Therefore, we have performed data augmentation using SpecAugment to reduce overfitting. The proposed base model is a convolutional neural network consists of the Inception modules and three types of pooling methods after each convolution blocks. We have trained this base model with two initializers and four validation split methods to create various classifiers. In addition, it was able to improve the performance with Stacking ensemble, and the classification accuracy of the best model was 76.1% on the development dataset and 81.3% on the leaderboard dataset.

6. 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)

7. REFERENCES

[1] http://dcase.community/challenge2019/

[2] Y. Sakachita and M. Aono, “Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions,” *IEEE AASP Challenge on DCASE 2018 technical reports,* 2018.

[3] Y. Han, J. Park and K. Lee, “Convolutional Neural Networks with Binaural Representations and Background Subtraction for Acoustic Scene Classification,” *IEEE AASP Challenge on SCASE 2017 technical reports,* 2017,

[4] Q. Kong, Y. Cao, T. Iqbal, Y, Xu, W. Wang and M. Plumbley, “Cross-task learning for audio tagging, sound event detection and soatial localization: DCASE 2019 baseline systems,” *arXiv:1904.03476v3,* 2019.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," *in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR),* 2015.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the Inception architecture for computer vision," *in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR),* 2016.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, "Inception-v4, Inception-Resnet and the impact of residual connections on learning," *in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI),* 2017.

[8] 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.

[9] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *in Proceedings of the 3rd International Conference on Learning Representations (ICLR),* 2015.

[10] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Artificial Intelligence and Statistics (AISTATS),* 2010, pp. 249-256.

[11] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *2015 IEEE International Conference of Computer Vision*, 2015, pp. 1026-1034.