GIST_WISENETAI AUDIO TAGGER BASED ON CONCATENATED RESIDUAL NETWORK FOR DCASE 2018 CHALLENGE TASK 2

Technical Report

Nam Kyun Kim[1], *Jeong Hyeon Yang*[1], *Jeong Eun Lim*[2], *Jinsoo Park*[2],
Ji Hyun Park[2], *and Hong Kook Kim*[1,*]

[1] School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology, 123 Cheomdangwagi-ro, Gwangju 61005, Korea
{skarbs001, didrn43, hongkook}@gist.ac.kr

[2] Algorithm R&D Team
Hanhwa Techwin, 319-6 Pangyo-ro, Sungnam 13488, Korea
{je04.lim, jspark82, jihyun.park}@hanwha.com

ABSTRACT

In this report, we describe the method and performance of an acoustic event tagger applied to the Task 2 of the Detection and Classification of Acoustic Scene and Events 2018 (DCASE 2018) challenge, where the task evaluates systems for general-purpose audio tagging with an increased number of categories and using data with annotations of varying reliability. The proposed audio tagger, which is call GIST_WisenetAI and developed by the collaboration of GIST and Hanwha Techwin, is based on a concatenated residual network (ConResNet). In particular, the proposed ConResNet is composed of two types of convolutional neural network (CNN) residual networks (CNN-ResNet) such as a 2D CNN-ResNet and an 1D CNN-ResNet using a sequence of mel-frequency cepstrum coefficients (MFCCs) and their statistics, respectively, as input features. In order to improve the performance of audio tagging, *k* different ConResNets are trained using *k*-fold cross-validation, and then they are linearly combined to generate an ensemble classifier. In this task, 9,473 audio samples for training/validation are divided into 10 folds, and 9,400 audio sample are given for testing. Consequently, the proposed method provides the mean average precision up to top 3 (MAP@3) of 0.958, which is measured through the Kaggle platform.

***Index Terms*—**DCASE, WISENET, audio tagging, concatenated residual network, sound event classification

1. INTRODUCTION

Recently, a large amount of multimedia data based on user smart mobile devices has been created. Such user-created contents are potentially incredibly valuable resources that can be used for commercial purposes and research applications. To this end, content understanding methods are increasingly attracted for automatic data tagging, segmentation, and indexing. However, labeling a huge amount of data is required for developing a reliable understanding model. Since such manual labeling is time-consuming, it is very important to develop an automatic labeling algorithm for providing appropriate information on the understanding of audio-visual data so that content understanding methods can be implemented.

For understanding audio contents, several efforts have been made concerning audio-tagging tasks by using machine learning techniques, such as hidden Markov model [1], Gaussian mixture model [2], support vector machine [3], and random forest [4]. However, thanks to the advances in deep learning, deep neural network based approaches have been successfully deployed for audio-related challenges, such as acoustic scene classification, music genre classification, and audio event detection. In addition, in the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events 2016 (DCASE 2016 Challenge), the VGG classifier [5] was adopted to assign a single label to an audio clip containing many sound events. In DCASE 2017 Challenge, a convolutional recurrent neural network (CRNN)-based audio event detection method was proposed in [6], and an attention mechanism through the output of gated recurrent units (GRUs) from CNNs was also proposed in [7]. Also, residual networks (ResNets) [8] have been shown to achieve high accuracy in the vision and audio field, which employs residual learning that utilizes skip connections between layers. Among the successful approaches in audio data understanding, ResNet-based audio classification achieved the higher mean average precision (MAP) score than other CNN based architectures [9].

In this report, an audio tagging method is proposed for the Task 2 of DCASE 2018 Challenge, where this task evaluates a system for general purpose audio tagging with increased number of categories by using data with annotations of varying reliability. Especially, the proposed audio tagging method[†] is based on a concatenated residual network (ConResNet). In particular, the proposed ConResNet is composed of two types of convolutional neural network (CNN) residual networks (CNN-ResNet) such as a 2D

[*] This work was support in part by Hanhwa Techwin.

[†] It is developed by the collaboration of Gwangju Institute of Science and Technology (GIST) and Hanhwa Techwin and it is called GIST_WisenetAI.

CNN-ResNet and an 1D CNN-ResNet using a sequence of mel-frequency cepstrum coefficients (MFCCs) and their statistics, respectively, as input features. In order to improve the performance of audio tagging, *k* different ConResNets are trained using K-fold cross-validation, and then they are linearly combined to generate an ensemble classifier. In this task, 9,473 samples for training/validation are divided into *k*=10 folds, and 9,400 sample are given for testing. Consequently, the proposed method provides the MAP up to top 3 (MAP@3) of 0.958, which is measured through the Kaggle platform.

Following this introduction, Section 2 proposes proposed ConResNet-based audio-tagging method. Section 3 evaluates the mean average precision score of the proposed audio-tagging method and compares it with those of different methods such as the baseline constructed by 3 CNNs [10], 1D ResNet, and 2D ResNet. Finally, Section 4 concludes this report.

2. PROPOSED GIST_WISENETAI AUDIO-TAGGING METHOD

This section describes the ConResNet applied to DACSE 2018 Challenge. As shown in Fig. 1, the proposed ConResNet is mainly composed of four stages: 1) feature extraction, 2) preprocessing of features prior to applying CNN-ResNets, 3) 1D CNN-ResNets and 2D CNN-ResNets, and 4) concatenation of the outputs of CNN-ResNets. Each stage will be described in the following subsections.

2.1. Feature extraction

In the first stage of the proposed ConResNet, each audio sample is divided into frames of 46 ms in length with an 11 ms overlap, where the sampling rate is set to 44.1 kHz. Then, a 2,048-point short-time Fourier transform (STFT) is applied to each audio frame. After that, 40-deimensional mel-frequency cepstral coefficients (MFCCs) are obtained by applying 60 mel-filterbanks followed by a discrete cosine transform. Each 40-dimensional MFCCs are expanded by concatenating their delta and delta-delta, resulting in 120-dimensional MFFCs.

For 2D processing, MFCCs from 500 audio frames are concatenated together so that the input layer of 2D CNN is 120 × 500 2D neurons. On one hand, four different statistical features for each audio sample are extracted such as the mean, standard deviation, skewness, and median values of each MFCC, resulting in 480 1D neurons.

2.2. Preprocessing for CNN residual networks

Instead of directly applying CNN-ResNets, the input layer composed of 120 × 500 neurons is connected into a 2D CNN, where 7×7 kernels are used and the number of strides is set to 2. Then, a 3×3 max pooling with stride 2 is performed. Similarly, the input layer composed of 480 neurons is connected into an 1D CNN, where 9 kernels are used with stride 2. Also, a 4 max pooling with stride 2 is performed

2.3. CNN residual networks

CNN-based networks have been popular because their performance is better than other architectures. However, they have problems in that they are hard to converge and have larger memory to train. Thus, ResNets have been proposed to train very deep CNNs [8]. ResNets are block-wise stacked architectures of the same shape, and each block in ResNets contains direct connections between the output of a lower layer and the inputs of a higher layer.

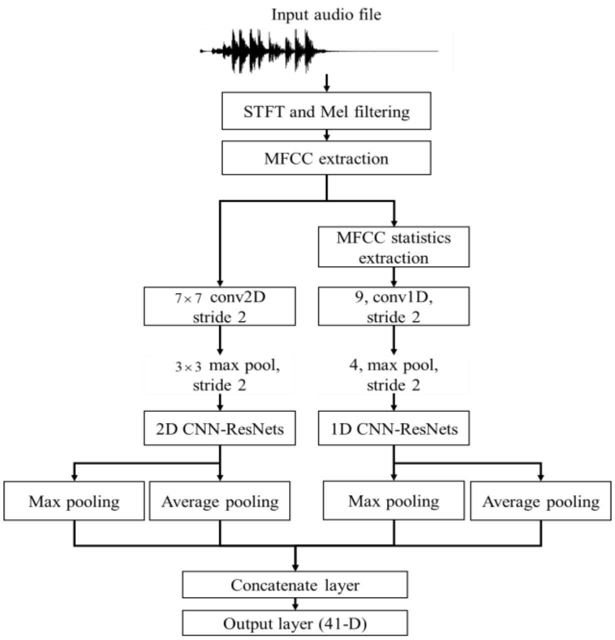

Figure 1: Procedure of the proposed ConResNet for audio tagging.

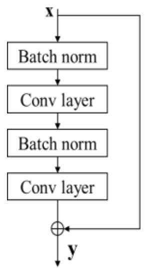

Figure 1: Structure of a residual unit used for the proposed ConResNet.

The 2D CNN-ResNets and 1D CNN-ResNets in the proposed ConResNet consist of four residual blocks each. In addition, each residual block is realized by concatenating several residual units. Fig. 2 shows a procedure of a residual unit, which computes the following equation of

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \mathbf{W}_i) + \mathbf{x} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors of the residual block, respectively, and $\mathcal{F}$ is a function that is composed of batch normalization and convolutional layers. In (1), $\mathbf{W}_i$ is the weights of the *i*-th residual unit.

Table 1 describes detailed architecture of the residual blocks used in the proposed ConResNet. As shown in the table, the first residual block of 2D CNN-ResNets is composed of three residual units described in Fig. 2. Each residual unit in the first residual

Table 1: Network architecture of the residual blocks used in the proposed ConResNet.

ResNet	2D CNN-ResNet	1D CNN-ResNet
Residual Block 1	$\begin{bmatrix}3\times3 & 32\\3\times3 & 32\end{bmatrix}\times 3$	$\begin{bmatrix}3 & 32\\3 & 32\end{bmatrix}\times 3$
Residual Block 2	$\begin{bmatrix}3\times3 & 64\\3\times3 & 64\end{bmatrix}\times 4$	$\begin{bmatrix}3 & 64\\3 & 64\end{bmatrix}\times 4$
Residual Block 3	$\begin{bmatrix}3\times3 & 128\\3\times3 & 128\end{bmatrix}\times 6$	$\begin{bmatrix}3 & 128\\3 & 128\end{bmatrix}\times 6$
Residual Block 4	$\begin{bmatrix}3\times3 & 256\\3\times3 & 256\end{bmatrix}\times 3$	$\begin{bmatrix}3 & 256\\3 & 256\end{bmatrix}\times 3$

block has two convolutional layers with 3×3 kernels. The second, third, and fourth residual blocks of 2D CNN-ResNets have 4, 6, and 3 residual units, respectively, and they all have two convolutional layers with 3×3 kernels, and the number of filter are 64, 128 and 256 respectively. The second column of Table 1 describes the architectures of residual blocks for the 1D CNN-ResNets. The architectures in the table are experimentally set by taking into account a trade-off between performance and complexity.

2.4. Concatenate layer and ensemble classifier

After applying the 1D CNN-ResNets and 2D CNN-ResNets, the outputs of both ResNets are processed by average pooling and max pooling, as shown in Fig. 1. Then, such four pooled layers are concatenated to identify 41 audio classes. Furthermore, in order to improve the performance of audio tagging, ten different ConResNets are trained by using 10-fold cross-validation from the training and validation data. Finally, they are linearly combined to generate an ensemble classifier.

3. PERFORMANCE EVALUATION

AudioSet [11] introduces the structured hierarchical ontology of 632 audio classes, which guides the literature and manual curation. This task employed audio samples from Freesound [12] annotated using a vocabulary of 41 labels from Google's AudioSet ontology. Each audio sample in this task was approximately 300 ms to 30 s long and sampled at a rate of 44.1 kHz. The training data set consisted of 3,710 manually verified annotations and 5,713 non-verified annotations that were roughly estimated to be 70% in each sound event.

The training data were augmented by stretching, shifting frames, and employing additive white Gaussian noise. Furthermore, the proposed ConResNet was trained with the mini-batch ADAM optimization algorithm to minimize the categorical cross-entropy criterion. The training data was divided into 10 folds. Each fold was then used once as a validation, while the nine remaining folds were used for training. Finally, an ensemble classifier was obtained by linearly combining 10 ConResNets, which was called GIST_WisenetAI.

The performance of the proposed GIST_WisenetAI was evaluated by measuring the mean average precision up to top 3 (MAP @ 3). MAP@3 was defined by

Table 2: Comparison of MAP@3 scores between different methods applied to the evaluation set through Kaggle platform.

Method	MAP(@3)
Baseline (3 CNNs)	0.704
1D CNN-ResNet (statistics of MFCCs)	0.872
2D CNN-ResNet (2D MFCC features)	0.935
Proposed ConResNet	**0.958**

$$\mathrm{MAP@3} = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{\min(n,3)} P_u(k) \qquad (2)$$

where U is the total number of audio samples (=9,400 in this report), and $P_u(k)$ is the precision of the *u*-th sample at cut-off *k*, and *n* is the number of predictions per audio sample.

Table 2 compares MAP@3 scores between different methods applied to the evaluation set DCASE 2018 Challenge Task 2 through the Kaggle platform. The compared methods were 3 CNNs [10], 1D ResNet, and 2D ResNet. As shown in the table, the MAP@3 score of the proposed method was the highest among all the compared methods. In particular, the proposed method improved the MAP@3 score by 0.086 and 0.023, compared with the 1D CNN-ResNet and 2D CNN-ResNet, respectively.

4. CONCLUSION

This report proposed an audio tagging method by concatenating 1D ResNets and 2D Resnets, which was referred to as ConResNet. In addition, the proposed method constructed an ensemble classifier through the *k*-fold cross-validation. Then, the proposed method was applied to the Task 2 of the DCASE 2018 Challenge. We actively participated in this challenge through the Kaggle platform. As a result, we achieved the MAP@3 of 0.958.

5. REFERENCES

[1] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proc. 18th European Signal Processing Conference (EUSIPCO)*, 2010, pp. 1267–1271.

[2] http://www.cs.tut.fi/sgn/arg/dcase2016/task-audiotagging

[3] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. 24th ACM on Multimedia Conference*, 2016, pp. 1038–1047.

[4] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.

[5] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," Tech. Rep., DCASE 2016 Challenge, Sept. 2016.

[6] H. Lim, J. Park, K. Lee, and Y. Han, "Rare sound event detection using 1D convolutional recurrent neural networks," Tech. Rep., DCASE 2017 Challenge, Sept. 2017.

[7] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley. "Surrey-CVSSP system for DCASE2017 Challenge Task4," Tech. Rep., DCASE 2017 Challenge, Sept. 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[9] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, “CNN architectures for large-scale audio classification,” arXiv:1609.09430, 2016.

[10] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pon s, and X. Serra, “General-purpose tagging of freesound audi o with audioset labels: Task description, dataset, and baselin e,” arXiv preprint arXiv:1807.09902, 2018.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: an ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP*, 2017, pp. 776–780.

[12] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra. “Freesound datasets: a platform for the creation of open audio datasets,” in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 486–493.