END-TO-END DEEP CONVOLUTIONAL NEURAL NETWORK WITH MULTI-SCALE STRUCTURE FOR WEAKLY LABELED SOUND EVENT DETECTION

Technical Report

Seokjin Lee[1], *Minhan Kim*[1], *Youngho Jeong*[2]

[1] Kyungpook National University, School of Electronics Engineering, Daegu, Republic of Korea,
{sjlee6, kmh7576}@knu.ac.kr
[2] Electronics and Telecommunications Research Institute, Realistic AV Research Group,
Daejeon, Republic of Korea

ABSTRACT

In this paper, an end-to-end sound event detection algorithm that detects and classifies the sound events from the waveform itself. The proposed model consists of multi-scale time frames and networks to handle both short and long signal characteristics; the frame slides 0.1 second to provide sufficiently fine resolution. The element network for each time frame data consists of several one-dimensional convolutional neural networks (1D-CNNs) with deeply stacked structure. The results of element networks are averaged and gated by sound activity detection. The decision is made by performing the double thresholding, and the results are enhanced by class-wise minimum gap/length compensation. To evaluate our proposed network, the simulation was performed with development data from DCASE 2019 Task 4, and the results show that the proposed algorithm has a macro-averaged f1-score of 31.7% for the development dataset of DCASE 2019 and 30.2% for the evaluation dataset of DCASE 2018.

Index Terms— Sound event detection, end-to-end, convolutional neural network, raw waveform

1. INTRODUCTION

In order to utilize a machine to help people's daily lives, the researches that makes the machine understand the environment are important. Recently, several machine learning algorithms have been researched to deal with the task to understand the circumstance from a sound signal including acoustic scene classification (ASC)[1] and sound event detection problems (SED)[2].

Most of the machine learning algorithms for acoustic signals utilize frequency-domain features such as Mel-frequency cepstral coefficient (MFCC), Mel-frequency spectrum [3, 4], or constant-Q Transform[5]. However, the feature extraction module needs some parameters to be tuned, e.g. number of bins, window length and hopsize for short-time Fourier transform, and frequency resolution, and the values of parameters may affect the performance of the algorithms. Furthermore, it is hard to decide which feature is best for a specific task among the various kinds of features.

In order to tackle the problem, we consider an end-to-end approach that utilizes the a law waveform itself as a feature for the sound event detection. Our work is inspired by the SampleCNN[6], which is a deep convolutional neural network (CNN) architecture by stacking several one-dimensional (1D) CNNs, and enhance the model to be suitable to the SED task.

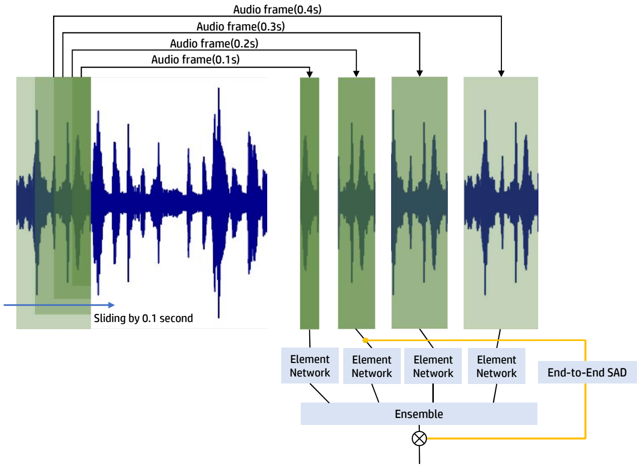

Figure 1: A block diagram for the proposed system.

2. PROPOSED MODEL

2.1. Overview for the architecture of the proposed sound event detector

In the SED task, not only the class but also the onset and offset information of each sound event have to be identified. Therefore, the temporal resolution of the prediction results have to be shorter than the target resolution, which is 0.2 seconds in the DCASE 2018 Task 4. In order to achieve this goal, the proposed SED system consists of multi-scale sliding temporal window as shown in Fig. 1. Every frames moves to the right by 0.1 second to provide sufficiently fine temporal resolution. Because each sound event class have different signal characteristics, the required signal length for each class is also different. In order to handle the both signal classes which have short and long signal characteristics, the proposed system utilize multi-scale windows such as 0.1, 0.2, 0.3, and 0.4 seconds. As shown in Fig. 1, the data from each time frame is fed to an element network.

2.2. Element networks

The data from a single time frame is fed as an input signal to the element network, as shown in Fig. 2. The element network consists

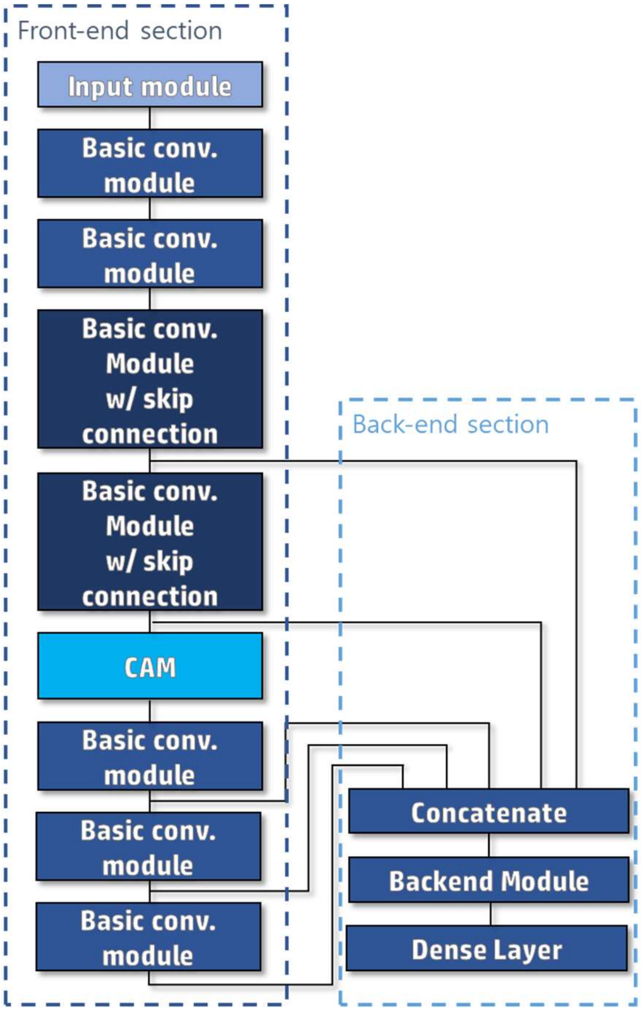

Figure 2: A block diagram for the element network.

of input module, basic convolutional module with and without skip connection, channel attention module (CAM), backend module, and output dense layer. The part of structure from the input module to the last basic convolutional module, which is shown in the left column of Fig. 2, is called *front-end section*, and the other part of structure from the concatenate module to output dense layer, which is shown in the right column of Fig. 2, is called *back-end section*.

Fig. 3 shows the specifications of the modules in the element network. The *front-end section* consists of several 1D-CNN layers and max-pooling layers as shown in the specification of basic convolutional module (top right of Fig. 3), because it is inspired by SampleCNN[6]. The third and fourth basic convolutional modules are modified with skip connection (middle of Fig. 3), which is inspired by ResNet[7] , to prevent performance degradation caused by deep architecture. In addition, a channel attention module (CAM) (bottom left of Fig. 3) is inserted between the fourth and fifth basic convolutional modules, which is inspired by convolutional block attention module (CBAM)[8], to realize a channel-wise attention. Two dense layers in the CAM have the activations of sigmoid functions, and they compose the bottle neck structure. The input module (top left of Fig. 3) consists of a 1D-CNN and the batch normalization, and the ReLU activation layer, and the backend module (bottom right if Fig. 3) is consists of a 1D-CNN with the ReLU activation and dropout, global max-pooling layer, and an attention module consists of a dense layer with the sigmoid activation and multiply.

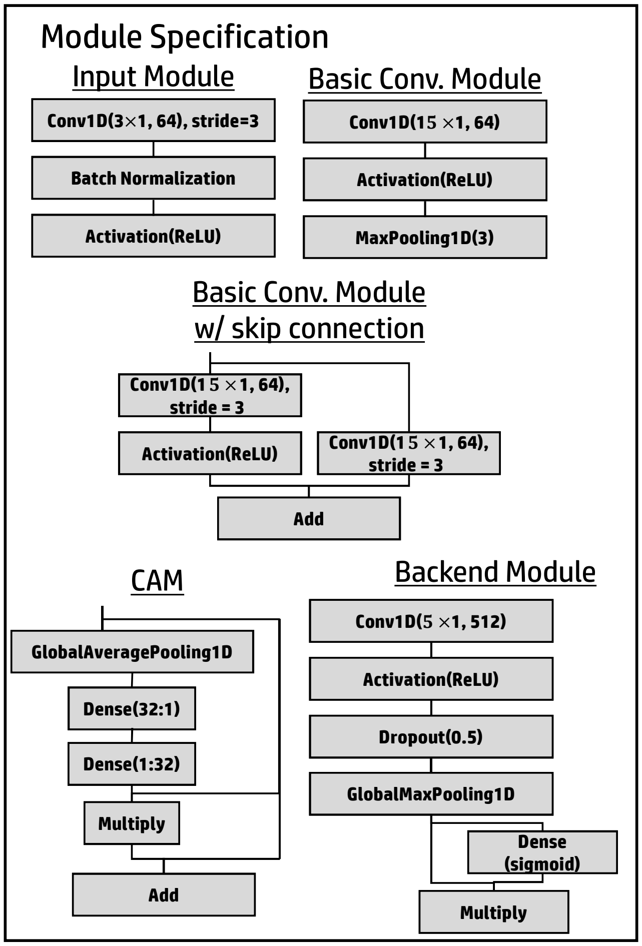

Figure 3: Specifications for network modules of the element network.

2.3. Ensembling and gating the outputs of the element networks

As shown in Fig. 1, the outputs of element networks are ensembled and gated by the end-to-end sound activity detection (SAD) module. The majority vote and averaging method have been tested as the ensembling method of the output signals, and the averaging method has been chosen because it shows more stable performance.

The outputs are multiplied by the prediction result of the end-to-end SAD module. The end-to-end SAD module is constructed as shown in Table 1, which is inspired by vadnet[9]. The SAD module is trained with the strongly labeled (synthetic) dataset where the ground truth label $\mathbf{y}_{SAD}(k)$ for k-th data is generated by the onehot-encoding as

$$\mathbf{y}_{SAD}(k) = \begin{cases} [1 \quad 0] & \text{if } \forall y \in \mathbf{y}_{strong}(k) = 0, \\ [0 \quad 1] & \text{if } \exists y \in \mathbf{y}_{strong}(k) = 1, \end{cases} \tag{1}$$

where $\mathbf{y}_{strong}(k)$ is a many-hot-encoding vector for k-th data of strongly labeled dataset. The prediction of the SAD module is thresholded by a predetermined value, then the outputs of the element networks are multiplied by the 2nd column of the prediction output matrix.

2.4. Post processing

There are two post processing methods applied to the proposed system: double thresholding and minimum gap/length compensation. The existence of each sound event class is estimated by thresholding the prediction result with high threshold value first, then the onset and offset is extended by searching the adjacent value below the low threshold. Then, we fill the gaps which are shorter than a predetermined minimum gap parameter, and remove the events which are shorter than a predetermined minimum length parameter.

2.5. Training Procedure

In order to handle the weakly labeled and unlabeled data, the training procedure consists of three stages:

- Train the model with strongly and weakly labeled data;
- Perform the transfer learning to the back-end section of the model with unlabeled data using the mean-teacher model[10];
- Tuning the weights of the back-end section using augmented data from the strongly and weakly labeled data (optional).

Because the weakly labeled data does not contain the onset and offset data, so the data was annotated by the SAD module in advance. The mean-teacher model is adopted from the previous research [10] and modified slightly: applying the mean-teacher model for the frame-wise only because our system cannot support the clip-wise comparison, and providing the label of the unlabeled data generated by pseudo-labeling with the trained model in the first stage. The augmented data were generated by applying the notch filter to the strongly and weakly labeled; it is inspired by SpecAugment[11] but using the notch filter instead of frequency masking, because our model requires the waveform of the augmented data.

3. SIMULATION RESULTS

In order to evaluate the propose algorithm, simulations were performed with the DCASE2019 task4 development dataset. The dataset consists of 2045 strongly labeled (synthetic) files, 1578 weakly labeled files, and 14412 unlabeled files, with 44100-Hz sampling frequency. The threshold of the SAD module was set to 0.4, the high thresholds for double thresholding were the values between 0.4 and 0.7 that are chosen to make good performance for the validation data in the development dataset, and the low thresholds for double thresholding were set to the average prediction values for each class. The length of minimum gap/length compensation was set to 1 second and it was applied to a long signal classes e.g. frying, blender, running water, vacuum cleaner, and electric shaver. The input data of test signals were noise-level-normalized to compensate the system gain difference.

The optimization method was set to Adam [12] with the step size of 10^{-3} in the first stage, stochastic gradient descent (SGD) with the step size of 10^{-3} in the second stage, and SGD with the step size of 10^{-4} in the third stage. The weight of the consistency loss of the mean-teacher model in the second stage was set to 10.0.

Table 2 shows the simulation results for the validation data of DCASE2019 and the evaluation data of DCASE2018 with applying various post processing methods. The performance measures are macro-averaged f1-score. The meaning of each name in Table 2 as follows.

- FixThr: double thresholding with fixed high threshold values;
- FixGap: double thresholding with fixed high threshold values and minimum gap/length compensation;
- VarThr: double thresholding with various high threshold values for each class;
- VarGap: double thresholding with various high thresholds and mininum gap/length compensation;
- FixGapAug: double thresholding with fixed high threshold values, minimum gap/length compensation, and data augmentation;
- VarGapAug: double thresholding with various high threshold values, minimum gap/length compensation, and data augmentation.

The results show that the propose algorithm with the post processing of double thresholding with various high thresholds and compensating the minimum gap/length has best performance among the various post processing methods. The segment-based metric of the proposed algorithm is slightly worse than or similar to the baseline performance, but the event-based metric of the proposed algorithm is better than the baseline performance for both of 2019 validation and 2018 evaluation dataset.

Table 1: Structure of SAD module

Layers
Conv1D(16 × 1, 64, strides = 2, activation = ReLU)
MaxPooling1D(4, strides = 4)
Conv1D(32 × 1, 32, strides = 2, activation = ReLU)
MaxPooling1D(4, strides = 4)
Conv1D(64 × 1, 16, strides = 2, activation = ReLU)
MaxPooling1D(4, strides = 4)
GRU(64, activation = tanh)
GRU(64, activation = tanh)
Dense(2, activation = softmax)

Table 2: Simulation results for various post processing methods with f1-score (2019 validation / 2018 evaluation)

Post processing	Segment-based	Event-based
FixThr	48.3 / 46.4	26.8 / 24.7
FixGap	49.9 / 46.3	29.3 / 27.6
VarThr	54 / 52.1	28.8 / 27.0
VarGap	**54 / 52.1**	**31.7 / 30.2**
FixGapAug	50.4 / 48.4	29.6 / 27.8
VarGapAug	52.1 / 50.1	30.6 / 29.2
Baseline	55.2 / 51.4	23.7 / 20.6

4. CONCLUSION

In this paper, an end-to-end sound event detection method based on deep CNN structure is proposed. The proposed algorithm consists of the front-end and back-end section: the front-end section substituting the feature extraction is developed based on deeply stacking the 1D-CNN layers; the back-end section is developed based on concatenating the outputs from intermediate stages of the front-end section, 1D-CNN and global max-pooling layers, and a self-attention module. To evaluate the proposed system, simulations with validation dataset from the DCASE2019 task4 and evaluation dataset from the DCASE 2018 task4 were performed to the proposed model trained by the DCASE2019 task4 training dataset. The simulation results show that the proposed algorithm has enhanced performance than the baseline measured by the event-based f1-score.

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] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, “Acoustic scene classification: Classifying environments from the sounds they produce,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, “Polyphonic sound event detection using multi label deep neural networks,” in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[3] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. J. Mataric, “Where am i? scene recognition for mobile robots using audio features,” in *2006 IEEE International conference on multimedia and expo*. IEEE, 2006, pp. 885–888.

[4] J.-J. Aucouturier, B. Defreville, and F. Pachet, “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[5] R. Raj, S. Waldekar, and G. Saha, “Large-scale weakly labelled semi-supervised cqt based sound event detection in domestic environments,” DCASE2018 Challenge, Tech. Rep., September 2018.

[6] J. Lee, J. Park, K. Kim, and J. Nam, “Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification,” *Applied Sciences*, vol. 8, no. 1, p. 150, 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[9] J. Wagner, D. Schiller, A. Seiderer, and E. André, “Deep learning in paralinguistic recognition tasks: Are hand-crafted features still relevant?” in *Proceedings of Interspeech*, Hyderabad, India, 2018, pp. 147–151.

[10] L. JiaKai, “Mean teacher convolution system for dcase 2018 task 4,” DCASE2018 Challenge, Tech. Rep., September 2018.

[11] 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.

[12] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.