SOUND EVENT DETECTION SYSTEM WITH MULTISCALE CHANNEL ATTENTION AND MULTIPLE CONSISTENCY TRAINING FOR DCASE 2022 TASK 4

Technical Report

Yu-Han Cheng, Chung-Li Lu, Bo-Cheng Chan, Hsiang-Feng Chuang
Chunghwa Telecom Laboratories, Taiwan, {henacheng, chungli, cbc, gotop}@cht.com.tw

ABSTRACT

In this technical report, we describe our submission system for DCASE 2022 Task4: sound event detection and separation in domestic environments. The proposed system is based on mean-teacher framework of semi-supervised learning and neural networks of CRNN. We employ consistency training of interpolation (ICT), shift (SCT), and clip-level (CCT) to enhance the generalization and representation. A multiscale CNN block is applied to extract various features to mitigate the influence of the event length diversity for the network. An efficient channel attention network (ECA-Net) and attention pooling enable the model to obtain definite sound event predictions. To further improve the performance, we use data augmentation including mixup, time shift, and filter augmentation. Our best system achieves the PSDS-scenario1 of 36.20% and PSDS-scenario2 of 63.45% on the validation set, significantly outperforming that of the baseline score of 32.93% and 53.22%, respectively.

Index Terms— sound event detection, CRNN, semi-supervised learning, consistency training, mean-teacher model, channel attention, pooling function

1. INTRODUCTION

This technical report describes our submission system for DCASE 2022 Task4: Sound Event Detection (SED) and separation in domestic environments. The goal of this task is to build a SED system to detect sound events and time boundaries in Scenario 1 (react fast) and Scenario 2 (avoid class confusion) by using a large amount of weakly labeled and unlabeled data. In this task, we employ a neural network and multiple strategies as below (Figure 1):

- CRNN [1] model.
- Multiscale CNN blocks [2] to extract various features.
- Consistency training of interpolation (ICT) [3], shift (SCT) [4], and clip-level (CCT) [5] to enhance model robustness.
- Efficient channel attention network (ECA-Net) [6] to pay more attention to important features.

To further improve the performance, we implement:

- Data augmentation methods including mixup [7], time shift, and filter augmentation [8] to increase data diversity.
- Adaptive post-processing to effectively smooth network output.

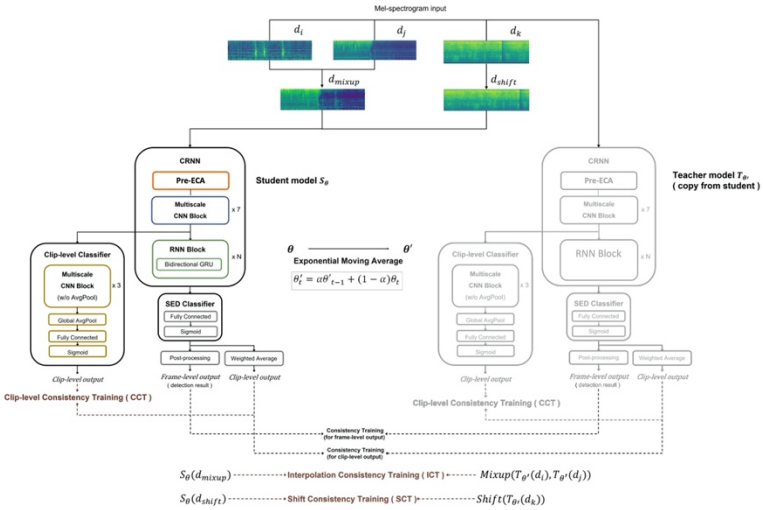


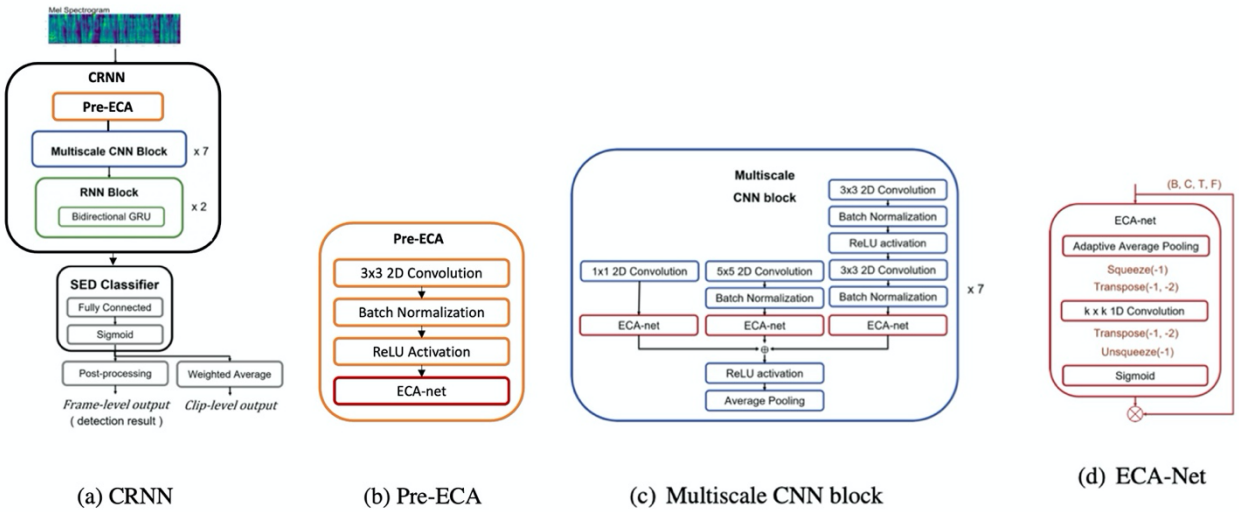
Figure 1: The proposed sound event detection system structure.

2. PROPOSED METHODS

2.1. Network architecture

2.1.1. CRNN

The convolutional recurrent neural network (CRNN) is similar to DCASE 2022 Task4 baseline architecture, which consists of 7 layers of CNN blocks and 2 layers of bidirectional gated recurrent unit (GRU), as shown in Figure 2(a). We try to add pre-ECA layer before the CNN blocks as a way to preprocess the input, as shown in Figure 2(b). A CNN block contains the convolutional layer, batch normalization (BN), Rectified Linear Unit (ReLU) activation, and average-pooling (AvgPool) layer. The input mel-spectrogram passes learnable convolution kernels and output the feature maps. BN and ReLU activation are intended to speed up and stabilize training. AvgPool calculates the average for each patch of the feature map and downsamples feature dimensions along both the time axis and the frequency axis. Then, RNN layers capture the long-term



(a) CRNN (b) Pre-ECA (c) Multiscale CNN block (d) ECA-Net

Figure 2: The network structure of CRNN, pre-ECA, multiscale CNN block, and efficient channel attention network (ECA-Net)

contextual information. Finally, the SED classifier consists of a fully connected layer and sigmoid function to discriminate the sound event types.

2.1.2. Multiscale CNN

From strongly labeled training data, we estimate duration of each sound event as below. 0~2s: alarm/bell/ringing, cat, dishes, dog, and speech. 4~6s: blender and running water. 7~10s: electric shaver/toothbrush, frying, and vacuum cleaner. The length of sound events is various and cause the model to work with inconsistent accuracy for the event of different scales. Thus, we refer to [2] to apply different kernel sizes to build a multiscale CNN block to capture the richer features, as Figure 2(c). A multiscale CNN block contains the kernel size of 1x1, 3x3, 5x5 and uses addition to integrate features of different scales.

2.1.3. Efficient Channel Attention

The effect of the acoustic feature extraction largely determines the model ability to predict different sound events and affects the final classification result. However, the attention mechanism can make the model pay more attention to areas which may be important features, and improve the model ability to distinguish features of sound events. We combine the efficient channel attention network (ECA-Net) [6] in multiscale CNN blocks before adding features of different scales, as shown in Figure 2(c). ECA-Net is composed of adaptive average pooling (A-AvgPool) layer, 1D convolutional (1D-CNN) layer, and sigmoid function, as shown in Figure 2(d).

A-Avgpool is applied along the channel axis and 1D-CNN calculate the attention of each channel. The kernel size of 1D-CNN is defined by

$$k = \left| \frac{\log_2(C) + b}{\gamma} \right|_{odd} \quad (1)$$

where k and C denote kernel size and channel dimension, γ and b are set to 2. Clearly, high-dimensional channels have longer range interaction, vice versa.

2.1.4. Pooling Function

[9] compared five different types of pooling functions in the multiple instance learning (MIL) framework for SED, namely attention pooling, max pooling, average pooling, linear softmax, and exponential softmax. The attention pooling estimates the weights for each frame are learned with a dense layer in the network. The max pooling simply takes the large probability in all frames. The average pooling assigns an equal weight for all frames. The linear softmax assigns weights equal to the frame-level probability, while the exponential softmax assigns a weight of exponential to the frame-level probability. We use attention pooling to transform frame-level into clip-level.

2.2. Semi-Supervised Learning

In this work, we employ the mean-teacher framework [10] for semi-supervised learning and use the Mean Square Error (MSE) loss for the consistency cost. The MSE loss function is defined by

$$\mathrm{MSE}(y, \hat{y}) = (y - \hat{y})^2 \quad (2)$$

where y and $\hat{y}$ denote the target and the prediction, respectively. Following, we propose multiple consistency criteria to regularize/direct how the SED system should learn from unlabeled or weakly-labeled data.

2.2.1. Interpolation Consistency Training

Recently, the interpolation consistency training (ICT) [3] has been proposed for semi-supervised learning. ICT encourages the prediction at an interpolation of unlabeled data points to be consistent with the interpolation of the prediction at these data points. Learning from interpolation samples can help the model discriminate ambiguous samples to improve the generalization ability. We define the ICT loss function by

$$L_{ICT} = \mathrm{MSE}(S_\theta(\lambda d_i + (1-\lambda)d_j), \lambda T_{\theta'}(d_i) + (1-\lambda)T_{\theta'}(d_j)) \quad (3)$$

where S_θ and $T_{\theta'}$ denote a student model and a teacher model, d_i and d_j denote data points, and λ is randomly sampled from a Beta distribution.

2.2.2. Shift Consistency Training

Inspired by ICT, we consider time-shift as another way to enhance consistency. It is called shift consistency training (SCT), which is similar to the method proposed by [4]. We define the SCT loss function by

$$L_{SCT} = \mathrm{MSE}(S_\theta(\mathrm{shift}(d_k)), \mathrm{shift}(T_{\theta'}(d_k))) \quad (4)$$

SCT encourages the prediction of time-shift input to be consistent with time-shift prediction. In theory, it allows the model to learn shift-invariance and temporal localization of sound events.

2.2.3. Clip-level Consistency Training

In addition to ICT and SCT, we also implement clip-level consistency training (CCT) [5]. We define the CCT loss function by

$$L_{CCT} = \mathrm{MSE}(\mathrm{NN}(d_x), \mathrm{ClipLevel}(f_x)) \quad (5)$$

where NN (d_x) is the weighted average pooling of the CRNN frame-level network output of data d_x, and Clip Level(f_x) is obtained by feeding the feature map f_x of the final CNN block to a clip-level classifier. As shown in Figure, the clip-level classifier consists of 3 extra multiscale CNN blocks, a global average pooling, and a fully connected layer.

2.2.4. Overall Consistency Training

In summary, the overall loss is

$$L = L_0 + L_{ICT} + L_{SCT} + L_{CCT} \quad (6)$$

where L_0 denotes the loss without the proposed consistency.

2.3. Data Augmentation

- Mixup [7]. It mixes two randomly selected samples from the original training data and uses λ sampled from Beta distribution to control the strength of interpolation between two samples. The linear interpolation technique can enhance the data diversity and robustness of the network.
- Shift [11]. It shifts a feature sequence on the time axis, and overrun frames are concatenated with the opposite side of the sequence. The usage helps the network learn temporal localization information of the sound event.
- Filter Augmentation [8]. It randomly increases or decreases dB of frequency bands on log mel-spectrograms. The improved version of frequency masking helps the model to extract information from wider frequency regions.

2.4. Adaptive Post-Processing

The frame-level network output is further post-processed to become the final output. First, thresholding operation converts probabilistic outputs to binary outputs. Then, the binary output sequences are further smoothed by median filters to avoid spurious detection. As sound classes may have varying temporal characteristics, we untie median filter sizes in the post-processing of the different sound classes. Following [12], we search the median filter size from 1 to 51 in increments of 1 with data from DCASE 2022 Task 4.

3. EXPERIMENTS

3.1. Dataset and Signal Preprocessing

The DESED dataset of DCASE 2022 Task 4 is comprised of 10-sec audio clips and 10 classes of sound events. The data are in two domains: real data (44.1kHz) extracted from Audio Set [13] and synthetic data (16kHz) generated by Scaper [14]. Each audio clip can be strongly labeled with the sound events and their time boundaries annotated, weakly labeled with only the sound events annotated, or unlabeled without any annotation. All dataset is divided into 4 subsets: weakly labeled (1,578 clips), unlabeled (14,412 clips), strongly labeled (10,000 clips), and validation set (1,168 clips). Audio signals are resampled to 16kHz sampling rate at first by librosa tool [15]. From the resampled signals, 128-channel mel-spectrogram is extracted with window size of 2048 and hop size of 256. The mel-spectrogram of a clip is normalized to zero mean and unit variance. Consequently, the size of the input acoustic features to the deep neural network is 626 × 128.

3.2. Network Setting

The 7 layers of multiscale CNN blocks have the number of filters: [16, 32, 64, 128, 128, 128, 128] and pooling size: [[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]]. For ICT and mixup augmentation, the parameter λ is sampled from Beta (α, α) and α from 0.1 to 0.7 in increments of 0.1. For SCT and shift augmentation, we choose the amount of time-shift by sampling from a normal distribution with a zero mean and a standard deviation of 90. For filter augmentation, dB range and band number range are set to (-9, 9) and (2, 5) respectively.

4. RESULTS

We submit two systems. The main difference between them is that system 1 implements pre-ECA in CRNN network whereas system 2 does not contain this part. In addition, system 1 applies mixup and filter augmentation for data augmentation while system 2 uses mixup and shift method.

The performance of each system is measured in terms of PSDS 1, PSDS 2, and macro-averaged event-based F1 score (F1_event). Table 1 shows the scores on the validation set of each system. Compared to the baseline, system 1 and system 2 are both improved. System1/system 2 increases 3.3%/1.6% and 10.2%/6.5% in PSDS 1 and PSDS 2 respectively. As for the F1 score, we can get higher score of 45.6% in system 1.

Table 1: system performance on the validation set.

system	PSDS 1	PSDS 2	F1_event
system 1	0.362	0.634	0.456
system 2	0.345	0.597	0.435
2022 baseline	0.329	0.532	0.403

5. CONCLUSION

In this technical report, the proposed system is based on the neural network of CRNN, which is trained with the mean-teacher framework of semi-supervised learning using multiple consistency criteria. Among them, interpolation consistency training (ICT) helps the model discriminate the ambiguous samples to enhance the generalization ability, shift consistency training (SCT) assists the model to learn better temporal information, clip-level consistency training (CCT) promotes the model feature representation power. In additional, a multiscale CNN block is applied to extract richer features to alleviate the influence of the diversity of event length for the model. An efficient channel attention network (ECA-Net) and attention pooling assist model to obtain more definite sound event predictions. We employ the mixup, shift, and filter augmentation as data augmentation to further improve the model performance. Finally, our best sound event detection system achieves the PSDS-scenario 1 of 36.2% and PSDS-scenario 2 of 63.4% on the validation set, considerably outperforming that of the baseline score of 32.93% and 53.22%, respectively.

6. REFERENCES

[1] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," 2019.

[2] M.Tang,L.Guo,Y.Zhang,W.Yan,andQ.Zhao,"Multi-scale residual crnn with data augmentation for dcase 2020 task 4."

[3] V.Verma,K.Kawaguchi,A.Lamb,J.Kannala,Y.Bengio,and D. Lopez-Paz, "Interpolation consistency training for semi- supervised learning," arXiv preprint arXiv:1903.03825, 2019.

[4] C.-Y. Koh, Y.-S. Chen, Y.-W. Liu, and M. R. Bai, "Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural net- works," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 376–380.

[5] L. Yang, J. Hao, Z. Hou, and W. Peng, "Two-stage domain adaptation for sound event detection."

[6] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca- net: Efficient channel attention for deep convolutional neural networks, 2020 ieee," in CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," arXiv preprint arXiv:1710.09412, 2017.

[8] Hyeonuk Nam, Seong-Hu Kim, and Yong-Hwa Park,

"FilterAugment: An Acoustic Environmental Data Augmentation Method", arXiv:2110.03282, 2022.

[9] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 31–35.

[10] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets im- prove semi-supervised deep learning results," arXiv preprint arXiv:1703.01780, 2017.

[11] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data aug- mentation method for automatic speech recognition," arXiv preprint arXiv:1904.08779, 2019.

[12] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution augmented transformer for semi-supervised sound event detection," in Proc. Workshop

[13] J. F. Gemmeke, D. P. 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 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 776–780.

[14] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 86–90.

[15] librosa, “librosa”,
https://github.com/librosa/librosa,2020.