DCASE 2022 CHALLENGE TASK4 TECHNICAL REPORT

Technical Report

Minjun Chen[1], Tian Wang[2], Jun Shao[1], Yiqi Tang[2], Yangyang Liu[1], Bo Peng[1], Jie Chen[1], Xi Shao[2]
[1] Samsung Research China-Nanjing, Nanjing, China
{minjun.chen, jun.shao, yang17.liu, b.peng, ada.chen}@samsung.com
[2] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China, {1021010412, 1221013731, shaoxi}@njupt.edu.cn

ABSTRACT

We describe our submitted systems for DCASE2022 Task4 in this technical report: Sound Event Detection in Domestic Environments. We propose three models to solve this problem. In the first model, we try to utilize all the training data provided. To be specific, firstly, we employ a joint model both for event classification and location based on strongly labeled data and weakly labeled data to propagate the clip level annotations on the unlabeled dataset, which is so called pseudo-label dataset. In order to link frame level strongly annotations with the weakly annotations, we introduce weighted average pooling scheme. Finally, the joint model trained on strongly labeled data, weakly labeled data and pseudo-label data are employed to solve the Task 4 problem. To utilize the external dataset and pre-trained model, we proposal a system which use pre-trained model to extract embedding, and to train a RNN decode to generate prediction finally. And the third system with some data augmentation methods based on the baseline CRNN. Our proposed systems achieve poly-phonic sound event detection scores (PSDS-scores) of 0.4428 (PSDS1) and 0.8266 (PSDS-scenario2) respectively on development dataset.

***Index Terms*—**Sound event detection, Pseudo labels, CRNN, AST, Segmenter

1. INTRODUCTION

In this technical report, we describe our submitted systems for the task 4 of the DCASE2022 challenge: Sound Event Detection in Domestic Environments using weakly labeled data, unlabeled data, and strongly labeled data. The target of the system is used to provide not only the audio event class but also the event time localization given that multiple events can be presented in an audio recording [1].

Based on the experimental observations, we propose three schemes as following:

In the first model, we utilize all the training data provided. Firstly, we employ a joint model both for event classification and location based on strongly labeled data and weakly labeled data to propagate the clip level annotations on the unlabeled dataset, which is so called pseudo-label dataset. In order to link frame level strong annotations with the weak annotations, we introduce weighted average pooling scheme. Finally, the joint model is trained on strongly labeled data, weakly labeled data and pseudo-label data. The joint model is based on CNN14, and then connect two parallel fully connected layers, one for frame level event classification and the other for sound event time localization. To map strong labels at the frame level to weak labels at the clip level, we use weighted average pooling.

In the second model, we utilize the AST[2] pre-trained model as feature extractor to extract frame-wise embedding, and use 2 layers RNN and a linear layer to transform the patch embedding to frame-wise predictions. We use all the synthetic data, weakly labeled data, unlabeled data, and strongly real labeled data to train our model.

In the third model, based on the baseline CRNN, we applied data augmentation methods and tried different weights for strong and weak loss, and also used the AudioSet strong labeled data to train this model.

This technical report is organized as follows: Section 2 details the models and tricks we use to train the SED systems. In Section 3, we demonstrate the experimental results of our proposed scheme. Finally, we conclude in Section 4.

2. PROPOSED METHOD

2.1. Data

We train and validate the proposed models on the datasets provided by DCASE2022 task4:

• Weakly labeled training set: This set contains 1578 clips (2244 class occurrences) for which only provide audio event classes for audio clips.

• Unlabeled in domain training set: This set contains 14412 clips which is considerably larger than weakly labeled data.

• Synthetic strongly labeled set: This set is composed of 10000 clips generated with the Scaper soundscape synthesis and augmentation library.

• Validation set: The validation set which is annotated with strong labels, with timestamps contains 1168 clips (4093 events).

2.2. Feature

For the first model, we extract log-mel features with hop size of 160 and window length of 512, on the resampled 16kHz audio data. 64 mel-filters are applied to obtain the final frame-wise features. All the training audio are aligned to 1000 frames which corresponds to 10 seconds. We use BatchNorm2D to normalize all the samples in development set.

For the second model, we extract fbank features with 128 mel-bins, 10ms frame shift and 16000HZ sample rate.

For the third model, we extract log-mel features with hop size of 256 and window length of 2048 on the resampled 16kHz audio data. All the training audio are aligned to 156 frames which corresponds to 10 seconds.

2.3. Semi-supervised Strategy

Semi-supervised strategy is essential due to the large amount of unlabeled data. In the first model, we introduce a pseudo-labeling [4] method. We first pretrain a neural network with labeled datasets (strongly labeled data and weakly labeled data) to label large-scale unlabeled data with weak pseudo labels.

2.4. Models

Model based on CNN14: In this model, we first train a joint model both for event classification and location on strongly labeled data and weakly labeled data to propagate the clip level annotations on the unlabeled dataset. And then use strongly labeled data, weakly labeled data and pseudo-label data to train the joint model again. The block diagram is shown in Figure 1.

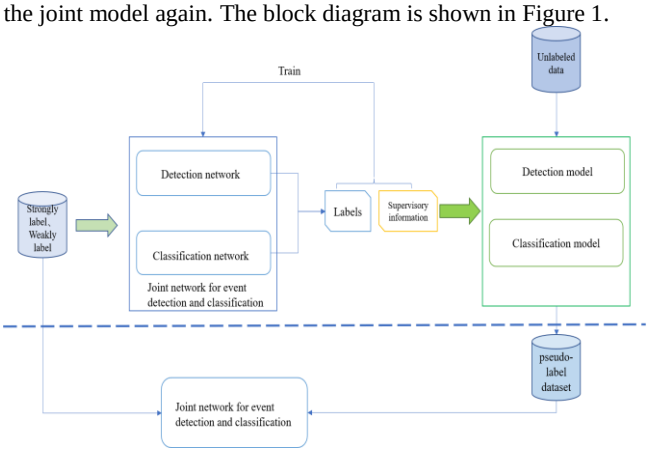

Figure 1: In the upper part of the figure, the joint model is pre-trained with labeled data, and the unlabeled data is labeled with pseudo labels. In the lower part, the original labeled data and pseudo label data are used to train the joint mode

We use Cnn14 as a feature extractor, then connect two full connection layers in parallel, one for frame level event classification and the other for sound event time localization, as shown in Figure 2.

We define the Fc-sigmoid output, $O_{soft}(t,i)$,as the localization vector, then it is multiplied with the classification output $O_{sig}(t,i)$ at each frame as,

$$O(t,i) = O_{sig}(t,i) \odot O_{soft}(t,i)$$

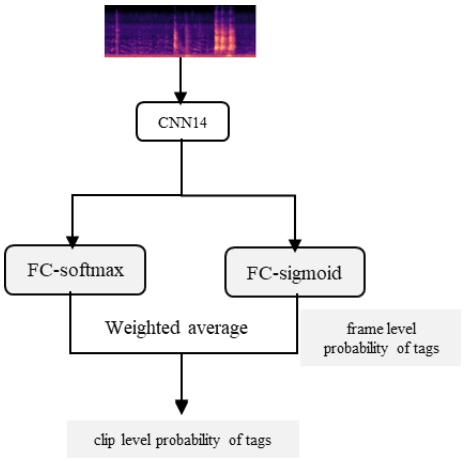

Figure 2: Model 1 Structure

Where $\odot$ means that the corresponding elements in the matrix are multiplied. t,i are the indexes of the time and events respectively. There are ten types of events in this task, so the index of i is 0 to 9. In order to map the frame level labels to the clip level strong labels, we introduce weighted average pooling (WAP) [5], it is defined as follows:

$$y_i = \frac{\sum_{t=0}^{T-1} O(t,i)}{\sum_{t=0}^{T-1} O_{soft}(t,i)} = \frac{\sum_{t=0}^{T-1} (O_{sig}(t,i) \odot O_{soft}(t,i))}{\sum_{t=0}^{T-1} O_{soft}(t,i)}$$

Where, y_i is the predicted weak label of the audio clip, and T is the total frames of an audio clip.

We calculate the binary cross entropy loss for strongly labeled data, weakly labeled data and weak pseudo label data, and take their sum as the loss function of the network. Since we introduce large-scale pseudo label data for training, the ratio between existing labeled data and pseudo label data has a great impact on the network performance [6]. Therefore, our loss function is defined as follows:

$$L = L_{strong} + L_{weak} + \alpha(t) L_{pseudo}$$

Where, $\alpha(t)$ is the balance coefficient between labelled data and pseudo label data. In this paper, we set it as 0.3. Model is trained with 200 epochs using Adam.

Model based on AST pre-trained model: In this mode, we use the AST[2] pre-trained model to extract the patch embeddings, then use a decoder to transform the embedding to frame-wise output as image segmenter[6]. The model diagram is shown in Figure 3.

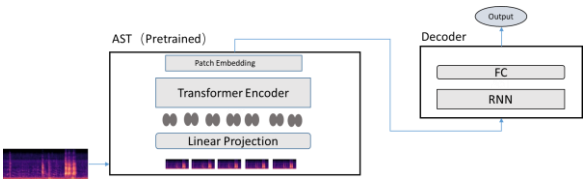

Figure 3: Model 2 Structure

The decoder uses two layers GRU, and then a linear layer to predict the frame-wise classes and the global clip-wise classes is generated by a weighted average of the frame-wise output.

Model based on CRNN: we use the popular convolutional recurrent neural net-work(CRNN). Traditional 2D convolution enforces translation-invariance in both time and frequency axis. However, frequency is not translation-invariant for sound time-frequency patterns. So we apply Frequency Dynamic Convolution[10] in our CNN. Also, we test the different weights for strong and weak loss. Apply different down-sampling rate for model optimization. We use all the synthetic data, weakly labeled data, unlabeled data, and strongly real labeled data (3377 audio clips coming from Audioset, total 3470, some of them are not successfully downloaded) to train this model.

2.5. Data augmentation

We apply different data augmentation in our models. we compared the effects of SpecAugment[7], include time warp, time and frequency masking, time-frequency shift[9] and mixup[8], and found the SpecAugment[7] does not have great improvements on our model based on pre-trained models. We finally used time-frequency shift[9] and mixup[8] in our models.

2.6. Post processing

We checked our predictions, and found it generate too many low-confidence predictions for a single clip. We try to apply different down-sampling methods to reduce the affect of the low-confidence predictions and find it could improve the PSDS2 scores.

3. EXPERIMENTS AND RESULT

For the decoder which uses 2 layers GRU, we compared the performance with LSTM:

Table 1: Performance of different RNN

	PSDS1	PSDS2
GRU	**0.3916**	0.7422
LSTM	0.3719	**0.7703**

As can be seen, LSTM get better PSDS2 but lower PSDS1.

For the different down-sampling methods during predictions post-processing, will improve PSDS 2 but different performance:

Table 2: Performance of different down-sampling methods

Frames	linear	maxpool
70	0.7762	0.7856
50	0.7842	0.7947
20	0.7788	0.8012
10	0.7702	0.8069
1	0.6598	**0.8266**

The final result of the systems we submitted are shown in Table 3. System 1 and 2 aim to better PSDS2, and system 3 is for PSDS1, system 4 is the one without external data.

Table 3: Performance of our proposed systems

id	system	PSDS1	PSDS2
1	AST + Segmenter 1	0.0670	**0.8266**
2	AST + Segmenter 2	0.1772	0.8012
3	CRNN	**0.4428**	0.6597
4	Detect + Classify	0.0375	0.2445

4. REFERENCES

[1] https://dcase.community/challenge2022/

[2] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv*:2104.01778, 2021.

[3] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[4] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018IEEE international conferenceon acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[5] Hou Yuanbo. Audio Tagging and Sound Event Detection based on Fuzzy Label, Master thesis, Beijing University of Posts &Telecommunications,2020

[6] Strudel, Robin and Garcia, Ricardo and Laptev, Ivan and Schmid, Cordelia, "Segmenter: Transformer for Semantic Segmentation" arXiv preprint arXiv:2105.05633 2021

[7] 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.

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

[9] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 Task 4," Tech. Rep. in DCASE 2019 Challenge, Orange Labs Lannion, France, 2019.

[10] Hyeonuk Nam, Seong-Hu Kim, Byeong-Yun Ko, Yong-Hwa Park "Frequency Dynamic Convolution: Frequency-Adaptive Pat-tern Recognition for Sound Event Detection" arXiv:2203.15296v1 2022.