DCASE 2023 CHALLENGE TASK4 TECHNICAL REPORT

Technical Report

Minjun Chen[1], Yongbin Jin[1], Jun Shao[1], Yangyang Liu[1], Bo Peng[1], Jie Chen[1]
Samsung Research China-Nanjing, Nanjing, China
{minjun.chen, yb.jin, jun.shao, yang17.liu, b.peng, ada.chen}@samsung.com

ABSTRACT

We describe our submitted systems for DCASE2023 Task4 in this technical report: Sound Event Detection with Weak Labels and Synthetic Soundscapes (Subtask A), and Sound Event Detection with Soft Labels (Subtask B). We focus on construct a CRNN model, which fuses the embedding extracted by the BEATs or AST pre-trained model，and use the frequency dynamic convolution(FDY-CRNN) and channel-wise selective kernel attention (SKA) for having adaptive receptive field. To get multiple models of different architectures for making an ensemble, we fine-tune multiple BEATs model on the SED dataset also. In order to make use of the weak labeled and unlabeled subset of DESED dataset further, we pseudo labels these subsets by a multiple iterative of self-training. We also use a small part of audio files from the Audioset dataset, and this part of data following the same self-training procedure. We train these models using two different settings, one setting for optimizing PSDS1 score, and the other for optimizing PSDS2 score. Our proposed systems achieve poly-phonic sound event detection scores (PSDS-scores) of 0.570 (PSDS-scenario1) and 0.889 (PSDS-scenario2) respectively on development dataset of subtask A, and macro-average F1 score with optimum threshold per class ($F1_{MO}$) 49.70 on development dataset of subtask B.

Index Terms**—**Sound event detection, Soft labels, Pseudo labels, CRNN, AST, BEATs, Self-training

1. INTRODUCTION

In this technical report, we describe our submitted systems for the task 4 of the DCASE2023 challenge [1]. The target of the subtask A is, give an audio clip, to predict not only the audio event class existing, but also the event onset and offset timestamps, given that multiple events can be presented in an audio recording. The target of the subtask B is to evaluate systems for the detection of sound events that use softly labeled data for training in addition to other types of data such as weakly labeled, unlabeled or strongly labeled. The focus of this subtask is to investigate whether using soft labels brings any improvement in performance.

It has been recognized that model pre-trained on large datasets could be transferred to downstream tasks, and bring significant performance improvement of downstream tasks. The BEATs [2] and AST [3] pre-trained models, trained on Audioset [4] dataset and other big datasets, have shown great progress on classification of sound events.

In addition, iterative self-training has been used widely in self-supervised learning tasks. For the subtask A, we mainly focus on using the pre-trained model to train multiple models of different architectures with the self-training method, and then make an ensemble from them as our final solution.

For subtask A, we train three different architectures of model. For the first model (SK-FD-CRNN), we use a CRNN to fuse the features extracted by BEAT pre-trained model, and we use the frequency dynamic convolution (FDY-CRNN) [5] and channel-wise selective kernel attention (SKA) [6] to replace the normal convolutions. For the second model (FT-BEATs), we fine-tune the pre-trained BEATs model directly using the DESED dataset, by adding a RNN layer and a linear layer as the prediction and output network. For the third model (FT-BEATs-AST), we fused AST embedding to fine-tune the BEATs, much like the second model (FT-BEATs). For all these models, we follow the same training procedure, and use an iterative self-training to pseudo labeling the weak labeled and unlabeled subset of SESED and Audioset dataset.

For subtask B, in the first model, in order to give an in-depth analysis of the class-wise performance for the under-represented classes, we applied data augmentation methods for the small classes. In the second model, we utilize the AST pre-trained model as feature extractor to extract frame-wise embedding, and use a linear layer to transform the patch embedding to frame-wise predictions.

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

2. PROPOSED METHOD

2.1. Data

We train and validate the proposed models on the datasets provided by DCASE2023 task4 (DESED) and the external dataset Audioset.

Subtask A:

- Weakly labeled training set: This subset contains 1578 clips (2244 class occurrences) for which only audio event classes (no timestamp) for audio clips are provided.
- Unlabeled in domain training set: This subset contains 14412 clips which are considerably larger than weakly labeled data, and not labels for this part of data.
- Synthetic strongly labeled set: This subset is composed of 10000 clips generated with the Scaper soundscape synthesis and augmentation library, strong labeled with timestamp of sound events.
- Validation set: The validation subset which is annotated with strong labels, contains 1168 clips (4093 events).

• Real strongly labeled training set: This part is composed of 3469 (total 3470, one of them is not downloaded successfully) audio clips coming from Audioset, strong labeled with timestamp of sound events. This part is considered as external dataset.
• Subset of Audioset: This part of data (32975 clips) is selected from Audioset by simple label mapping, we dropped the labels and used as external dataset.

Subtask B:
MAESTRO Real [7]: The dataset consists of real-life recordings with a length of approximately 3 minutes each, recorded in a few different acoustic scenes.

2.2. Feature

For subtask A, we used the feature, which is used during the training procedure of pre-trained models for fine-tune the proposed models. The BEATs pre-trained model use frame-shift of 10ms and window length of 25ms, 128 mel-bins, on the resampled 16 kHz audio data. The AST pre-trained model use 128 mel-bins and 10ms frame-shift. We use 128 mel-bins, 256 hop-length, 2048 window-length for training the SK-FD-CRNN network, which fuse the embedding extracted from BEATs and/or AST pre-trained models.
For subtask B, the first system is a CRNN with a linear output layer that is trained using the soft labels and MSE loss. As input, the model uses mel-band energies extracted using a hop length of 200ms and 64 mel filter banks. For the AST pre-trained model, we extract log-mel features with window length of 1024, on the resampled 16 kHz audio data. 128 mel-filters are applied to obtain the final frame-wise features. All the training audios are aligned to 10 seconds. We use BatchNorm2D to normalize all the samples in development set.

2.3. Iterative Self-training Strategy

Self-training strategy is widely used in visual/sound deep learning, due to the larger amount of unlabeled data than labeled data. In the subtask A, we pseudo label the weak labeled and unlabeled in domain subsets by using an iteratively procedure.
We first use all the SESED dataset and the Real strongly labeled subset (Audioset) to training multiple models of the proposed three architectures, and then make a big ensemble from these models for pseudo labeling the weak and unlabeled subset, and use a class-wise fine-tuned thresholds and median filter length as [8] when pseudo labeling. The pseudo labeled subsets are used to training/fine-tune the models iteratively. We finally make a big ensemble by selecting models, which are trained during the iterative self-training procedure as our final submitted systems. For the Subset of Audioset (32975 clips), we following the same self-training procedure. The iterative self-training is performed for two rounds.

2.4. Models

Subtask A proposes three models as following.

SK-FD-CRNN: In this model, we use the frequency dynamic convolution (FDY-CRNN) and selective kernel attention (SKA) to replace the normal convolutions of a 7-layers CNN network, and fuse the embedding extracted by BEATs pre-trained model. We use two methods (pool1d and interpolate) to fuse the embedding and outputs of CNN to train two different types of model, the fused features are then feed into a RNN classification network. The net- work is shown in Figure 1.

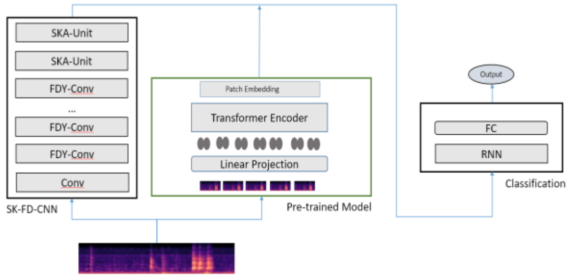

Figure 1: SK-FD-CRNN network structure, the output of CNN network and the embedding extracted by pre-trained model (BEAT) are fused and feed to a RNN output network for classification

FT-BEATs: in this model, we fine-tune the BEATs on the provided SESED dataset and Audioset subset, by adding a 4-layers Bi-GRU and a linear layer as the output network after BEATs. All the parameters are not frozen and be trained with a small learning rate.

FT-BEATs-AST: this model is similar with FT-BEATs, except than the AST pre-trained model is also participate in the fine-tune procedure, with parameters be updated also. The outputs from AST and BEATs are fused by pool averaging. The learning rate is set to a small (0.0001).

When training subtask A models, mean-teacher [9] semi-supervised structure is applied.

Subtask B propose AST pre-trained model as following.

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

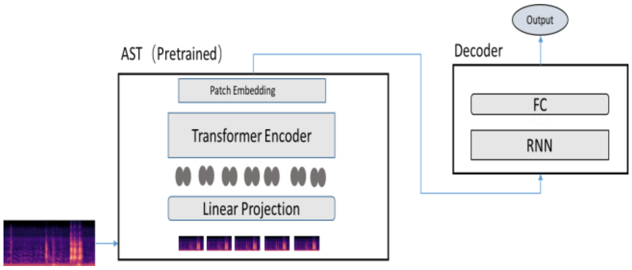

Figure 2: Subtask B Model Structure

The decoder uses one bi-directional gated recurrent unit (GRU) layer, then a linear layer to predict the frame-wise classes.

2.5. Data augmentation

In order to give an in-depth analysis of the class-wise performance of the under-represented classes for subtask B, we applied oversampling for the small classes. We compared the effects of mix-up and oversampling of small classes.

Table 1: Performance of data augmentation

	Micro-average		Macro-average	
	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
Base CRNN	0.474	69.8	35.05	43.05
mix-up	0.512	69.97	36.36	43.85
oversampling	0.552	68.93	36.06	43.49

Table 1 shows that mix-up has the best $F1_{MO}$ performance, oversampling of the small classes also improves macro-average performance.

2.6. Post processing

We use different median filter lengths for different sound event for subtask A and B. In addition, a tagging mask strategy is used to filter out the strong predictions as [8] for subtask A. We use two sets of different hyper-parameters to train two sets of models, the first set of hyper-parameters aims to get better PSDS2 [10][11] scores, the other set aims to get better PSDS1 sores. The first set of hyper-parameters use 39 frames, the second use 156 frames to give different of time resolution. The models trained with the first set of hyper-parameters are used to extract the tagging masks for post processing, which would be applied on the outputs of models trained with the second set of hyper-parameters.

3. EXPERIMENTS AND RESULT

3.1. Subtask A

We prepare five different systems for submission.
- System1: This system is an ensemble of 6 models, aims to get better PSDS1 scores, the output is 156 frames, and use the tagging masks extract by System2 to filter out outputs.
- System2: This system is an ensemble of 20 models, aims to get better PSDS2 scores, the output is 39 frames, and different median-filter lengths for different sound events are applied.
- System3: This system is an ensemble of 33 models, only use the fixed median filter length (7) for all the classes as the baseline [12], no other post-processing applied.
- System4: This system is single model (SK-FD-CRNN), trained without using external dataset or pre-trained models, and use the data augment methods as RCT [13], include mix-up, time-shift, time-mask, and frequency-mask.
- System5: This system is basic the same with System2, but the time sequence length is pooled to one frame for optimizing PSDS2 score.

The results of the systems we submitted on development dataset are shown in Table 2.

Table 2: Performance of subtask A proposed systems

id	system	PSDS1	PSDS2
1	System1	**0.570**	0.843
2	System2	0.414	0.884
3	System3	0.554	0.833
4	System4	0.436	0.675
5	System5	0.118	**0.889**

3.2. Subtask B

We prepare two different systems for submission.
- System1: This system is single model based on CRNN with data augment, trained without using external dataset or pre-trained models.
- System2: This system is trained with AST pre-trained model and ensemble with 40 models.

The results of the systems we submitted are shown in Table 3.

Table 3: Performance of subtask B proposed systems

	Micro-average		Macro-average	
	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
System1	0.50	71.04	34.73	43.98
System2	0.43	72.9	28.8	**49.70**

4. REFERENCES

[1] https://dcase.community/challenge2023/

[2] S. Chen, Y.Wu, C. Wang, S. Liu, Daniel Tompkins, Z. Chen, and Furu Wei, “BEATs: Audio Pre-Training with Acoustic Tokenizers,” arXiv preprint arXiv:2212.09058, 2022.

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

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

[5] 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.

[6] Xiang Li, Wenhai Wang, Xiaolin Hu, Jian Yang, "Selective Kernel Networks," in IEEE Conference on Computer Vision and Pattern Recognition, 2019

[7] Irene Mart ń-Morat ó and Annamaria Mesaros. Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31():902–914, 2023.

[8] J.Ebbers and R. Haeb-Umbach, "Pre-Training and Self-Training for Sound Event Detection in Domestic Environments", Technical Report for Challenge on Detection and Classification of Acoustic Scenes and Events 2022.

[9] Tarvainen A, Valpola H, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Advances in neural information processing systems, 2017, 30.

[10] Janek Ebbers, Reinhold Haeb-Umbach, and Romain Serizel. Threshold independent evaluation of sound event detection scores. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1021–1025. IEEE, 2022.

[11] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Metrics for polyphonic sound event detection", Applied Sciences, 6(6):162, 2016

[12] L. Delphin-Poulat and C. Plapous, “MEAN TEACHER WITH DATA AUGMENTATION FOR DCASE 2019 TASK 4,” technical report, dcase 2019.

[13] Nian Shao, Erfan Loweimi, and Xiaofei Li, “RCT: Random Consistency Training for Semi-supervised Sound Event Detection,” arXiv:2110.11144v3 2022.