SOUND EVENT DETECTION WITH HETEROGENEOUS TRAINING DATASET AND POTENTIALLY MISSING LABELS FOR DCASE 2024 TASK 4

Technical Report

Wei-Yu Chen, Chung-Li Lu, Hsiang-Feng Chuang, Yu-Han Cheng, Bo-Cheng Chan

Advanced Technology Laboratory, Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taiwan

{weiweichen, chungli, gotop, henacheng, cbc}@cht.com.tw

ABSTRACT

In this technical report, we briefly describe the system we designed for Detection and Classification of Acoustic Scenes and Events (DCASE) 2024 Challenge Task4: Sound Event Detection with Heterogeneous Training Dataset and Potentially Missing Labels. Our optimal single system employs a two-stage training process. Pretrained BEATs[1] model is utilized as front-end feature extractor, with Bi-GRU module as back-end classifier for each single frame. We employ the mean teacher method for semi-supervised learning, incorporating the EMA strategy to update parameters of the teacher model. Additionally, we generate pseudo-labels using the student model to leverage unlabeled data. For data augmentation, techniques such as mix-up and SpecAugment [2] are employed. Median filter is used for post-processing. The submitted system without ensemble, achieves a Polyphonic sound event detection scores-scenario 1 (PSDS1)[3] score of 0.50 and a mean partial AUC(mean pAUC) of 0.73, while with ensemble it achieves a PSDS1 score of 0.53 and a mean pAUC of 0.77 on the validation set.

***Index Terms*—**DCASE, sound event detection, mean teacher, pre-training

1. INTRODUCTION

The DCASE 2024 Task 4 competition is a continuation of the DCASE 2023 Task 4 extended task. This time, the organizers provide two different datasets with different levels of granularity in the file labels. Some of the classes are overlapped between two datasets, for which different evaluation metric are designed for comparing performance of different systems.

The DESED dataset consists of 10-second audio clips, which are recorded in home environments or synthesized using Scaper, focusing on 10 types of sound events. It has been used since DCASE 2020 Task 4. The MAESTRO Real dataset contains approximately 3-minute long real-life recordings from various acoustic scenes. The audio is annotated through Amazon Mechanical Turk, and soft labels are estimated based on the consensus of multiple annotators. Since the original training datasets have not been re-annotated, sound labels present in one dataset may exist but be unannotated in the other. Therefore, the system must handle potentially missing target labels during training. Additionally, there is some overlap in categories between the two datasets. The following categories have been merged: the 'People talking' label in MAESTRO is equivalent to the 'Speech' label in DESED, and the 'Cutlery & dishes' label in MAESTRO is equivalent to the 'Dishes' label in DESED.

The MAESTRO dataset provides 17 sound categories. Considering the confidence and quantity of soft label, , training and evaluation are conducted on the selected 11 categories of sound event: birds singing, car, people talking, footsteps, children voices, wind blowing, brakes squeaking, large vehicle, cutlery and dishes, metro approaching and metro leaving.

In this report, we describe our submission system for DCASE 2024 Task4. The content includes network architecture, data augmentation method, fusion strategy, post-processing method, and the ensemble result.

2. PROPOSED METHODS

2.1. Network architecture

2.1.1. Baseline architecture with varying parallel front-end feature extractor

The overall architecture is shown in Figure 1. We employs BEATSiter3+ for embedding extraction as front-end feature extractor, with models such as CRNN (the competition's baseline)[4], VGGSK[5], or FDYCRNN[6] parallelled to capture additional features during the training process. As for back-end, Bi-GRU is utilized for frame-level sound event classification.

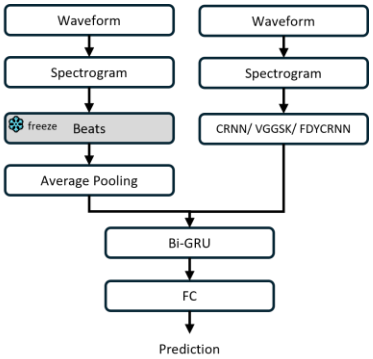

Figure 1: Overall architecture

2.1.2. Training strategy with mono pre-trained front-end

To minimize the complexity of experiments, we tried three training strategies with utilizing pretrained model BEATs solely as frontend, following with Bi-GRU as back-end.

1. Initializing with weights of pretrained BEATs, unfreezing and training with random initialization for the rest of the model.
2. Initializing with BEATs and freeze it, and training from scratch for the rest of the other part of the model. (Stage 1)
3. Initializing with model checkpoint of stage 1, fine tuning the whole model and unfreezing BEATs as well. (Stage 2)

Details of stage-1 and stage-2 are illustrated in Figure 2.

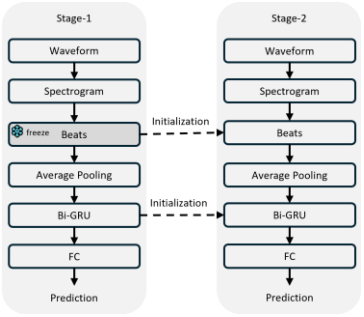

Figure 2: Training methods for pre-trained models in downstream tasks

2.2. Configuration Settings for Model Training

We employ the mean-teacher method for semi-supervised learning to train the identical teacher and student models with pseudo-labeled data. Binary Cross-Entropy (BCE) is used for supervised learning, and Mean Square Error (MSE) is employed for semi-supervised learning.

3. EXPERIMENTS

3.1. System Without Ensemble

3.1.1. Baseline Comparison

Table 1 illustrates the difference of the validation scores between the official annoucement and the results we reconstructed with official baseline model. We will use the results of the reconstructed baseline model as baseline for comparison in the following sections.

	PSDS1	mean pAUC
CRNN+BEATs (Official)	0.49	0.73
CRNN+BEATs (Baseline)	0.50	0.70

Table 1: We used the results generated by running the training model provided by the official source as our benchmark for comparison.

3.1.2. Experiment on baseline architecture with varying parallel front-end feature extractor

Table 2 primarily compares architectures utilizing different frontend. It can be observed that not significant difference for PSDS1, while FDYCRNN is slightly better-off regarding of mean pAUC.Thus, the BEATs model was separately trained and analyzed in the following section.

3.1.3. Experiment on training strategy with mono pre-trained front-end

Table 3 illustrates the performance of different training strategy described in 2.1.2 with pre-trained Beats as front-end.

1. It can be observed that directly fine-tuning the entire BEATs model for downstream tasks does not yield the best performance.
2. Freezing the BEATs component while training the downstream task achieves better results. For improving the PSDS1 performance.
3. Further training with Beats-stage2 can be applied, though the improvement is quite limited.

Based on the results from Tables 2 and 3, directly training the downstream task without ensemble achieves similar performance to the baseline in terms of PSDS1 and mean pAUC.

	PSDS1	mean pAUC
CRNN+BEATs (Baseline)	0.50	0.70
VGGSK+BEATs	0.49	0.69
FDYCRNN+BEATs	0.50	0.65

Table 2: Results of the single model within the CNN architecture + BEATs embedding on the validation set.

	PSDS1	mean pAUC
Beats	0.47	0.72
Beats-stage1	0.49	0.73
Beats-stage2	0.50	0.73

Table 3: Results of the single model within the BEATs architecture on the validation set.

3.1.4. Experiment on Post-Processing

From Tables 4 and 5, it can be seen that using the median filter for post-processing significantly improves the PSDS1 score, but has a limited effect on mean pAUC.

	Unprocessed	Post-processed
CRNN+BEATs(Baseline)	0.40	0.50 (+0.1)
VGGSK+BEATs	0.21	0.49 (+0.28)
FDYCRNN+BEATs	0.20	0.50 (+0.3)
Beats	0.20	0.47 (+0.27)
Beats-stage1	0.23	0.49 (+0.26)
Beats-stage2	0.39	0.50 (+0.11)

Table 4: PSDS1 results after comparing the post-processed outputs of each model.

	Unprocessed	Post-processed
CRNN+BEATs(Baseline)	0.70	0.70 (+0.0)
VGGSK+BEATs	0.69	0.69 (+0.0)
FDYCRNN+BEATs	0.65	0.65 (+0.0)
Beats	0.72	0.72 (+0.0)
Beats-stage1	0.72	0.73 (+0.01)
Beats-stage2	0.72	0.73 (+0.01)

Table 5: Mean pAUC Results after comparing the post-processed outputs of each model.

3.2. Results of Submitted Systems

Table 5 shows the results of our submitted systems on the validation dataset. System 1 uses the two-stage training BEATs. Systems 2, 3, and 4 select the best recognition categories either by averaging or by choosing the best category according to the candidate models. System 2 selects the most suitable ensemble method for DESED categories and MAESTRO categories separately. System 3 averages the candidate models. System 4 selects the best recognition category according to the candidate models. Among the four systems, the best results are achieved by System 2 and 4, with a PSDS1 of 0.53 and a mean pAUC of 0.77.

System	Ensemble	PSDS1	mean pAUC
CRNN+BEATs(Baseline)		0.50	0.70
System 1		0.50	0.73
System 2	✓	0.53	0.77
System 3	✓	0.53	0.74
System 4	✓	0.53	0.77

Table 6: Submitted Systems on validation set.

4. REFERENCES

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

[2] PARK, Daniel S., et al. Specaugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779, 2019.

[3] EBBERS, Janek; HAEB-UMBACH, Reinhold; SERIZEL, Romain. Threshold independent evaluation of sound event detection scores. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022. p. 1021-1025.

[4] S. J. Huang, C. C. Liu, C. P. Chen, C. L. Lu, B. C. Chan, Y. H. Cheng, H. F. Chuang, “CHT+ NSYSU SOUND EVENT DETECTION SYSTEM WITH DIFFERENT KINDS OF PRETRAINED MODELS FOR DCASE 2022 TASK 4. ” 2021

[5] https://github.com/DCASE-REPO/DESED_task

[6] LIU, Chia-Chuan, et al. Cht+ nsysu sound event detection system with pretrained embeddings extracted from beats model for dcase 2023 task 4. DCASE2023 Challenge, Tech. Rep, 2023.