Leveraging Semi-Supervised Learning with BEATs Feature Extraction and Bi-GRU Classification on Heterogeneous Datasets

Wei-Yu Chen[1], Chung Li Lu[1], Po Cheng Chan[1,2], Hsiang-Feng Chuang[1], Yu-Han Cheng[1], and Jia Ching Wang[2]

[1]Advanced Technology Laboratory, Chunghwa Telecom Laboratories, Taoyuan, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University
[1]Email: *weiweichen, chungli, cbc, gotop, henacheng@cht.com.tw*
[2]Email: *jcw@csie.ncu.edu.tw*

***Abstract*—In this paper, we present our approach for the DCASE 2024 Challenge, Task 4: Sound Event Detection in a Heterogeneous Training Dataset with Potentially Missing Labels. Our strategy employs a two-stage training protocol, initially using a pretrained BEATs model as the front-end feature extractor, coupled with a Bi-GRU module for frame-level classification. We adopt the mean teacher method, leveraging the EMA strategy to update the teacher model's parameters, and use pseudo labels from the student model to enhance the use of unlabeled data. Data augmentation techniques such as mix-up and SpecAugment are utilized, along with a median filter for post-processing. Our system, without ensemble methods, achieves a PSDS1 score of 0.50 and a mean partial AUC of 0.73. With ensemble techniques, performance improves to a PSDS1 of 0.53 and a mean pAUC of 0.77 on the validation set. Additionally, the final evaluation for a single model shows a PSDS of 0.495 on DESED and a mean pAUC of 0.733 on MAESTRO, while ensemble models reach a PSDS of 0.527 on DESED and a mean pAUC of 0.711 on MAESTRO.**

I. Introduction

Sound Event Detection (SED) aims to identify types of vocalization events and their start and end times within audio signals. SED has practical applications in areas like smart homes, traffic monitoring, and industrial production. Unlike the richly labeled datasets available in speech and image domains, audio datasets with strong labels are less prevalent. To augment the data pool for model training, methods such as unsupervised and semi-supervised learning can be employed to utilize a large volume of unlabeled samples.

The Convolutional Recurrent Neural Network (CRNN) [1] is a commonly used model architecture for Sound Event Detection (SED) systems. In this framework, the CNN module efficiently extracts local features from the feature map, while the RNN module processes temporal information in the audio signal, facilitating the extraction of contextually relevant features. Additionally, the Mean Teacher semi-supervised learning approach allows for the training of SED systems using both weakly labeled and unlabeled data.

The DESED dataset [2], includes 10-second audio clips recorded in residential environments or synthesized with Scaper, targeting 10 specific sound events. In contrast, the MAESTRO Real dataset contains longer, approximately 3-minute recordings from various acoustic settings. Annotations for MAESTRO [3] are gathered via Amazon Mechanical Turk, creating soft labels based on consensus among annotators. Given that certain sound labels appear in one dataset but not in the other, our system is designed to manage potentially missing labels during training. Additionally, there are overlapping categories between the datasets; for instance, MAESTRO's 'People talking' corresponds to the 'Speech' category in DESED, and 'Cutlery & dishes' in MAESTRO matches the 'Dishes' category in DESED.

II. Methodology

A. Pre-processing and Data Augmentation

All audio recordings are resampled to 16 kHz and converted to mono format. We use log-mel spectrograms, extracted using 128 mel bands, as acoustic features. These are derived from the Short-Time Fourier Transform (STFT), which is computed with a window size of 2048 and a hop length of 256, covering a frequency range from 0 Hz to 8000 Hz.

To increase our model's robustness, we employ data augmentation techniques such as mixup [4]a nd SpecAugment [5]. Mixup enhances generalization by blending pairs of audio samples and their labels to create new training instances, and is exclusively performed within the same dataset (e.g., only within MAESTRO or DESED). SpecAugment modifies the spectrograms directly.

B. Baseline

The model architecture is depicted in Figure 1. For the initial feature extraction, we use BEATs to obtain embeddings

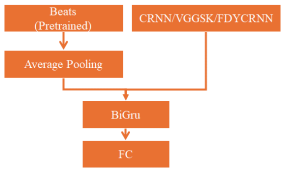

Fig. 1. Model Architecture

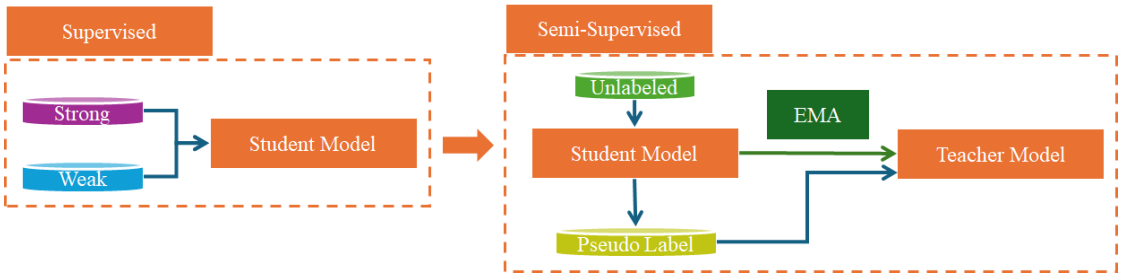

Fig. 2. Mean-Teacher Pipeline

as the front-end feature extractor. In parallel, models such as CRNN (the competition's baseline model) [1], VGGSK [6][7], and FDYCRNN [8] are employed to enhance feature capture during training. For the back-end processing, a Bi-GRU [9] model is implemented to classify sound events at the frame level.

C. Mean-Teacher

The mean-teacher algorithm [1][10] is used for semisupervised learning as shown in Figure 2.

The Sound Event Detection (SED) head composed of a fully connected layer, which outputs the frame-level prediction.

D. Mono Pre-trained Front-end Framework

The introduction of pre-trained models can greatly improve system performance. The BEATs [11] model achieved state-of-the-art scores in the Audioset [12] classification task. We implemented two training strategies, each utilizing the pretrained BEATs model as the front-end and Bi-GRU as the back-end.

1) Strategy 1: Monolithic Framework with Unfrozen Pre-trained Front-end: We initiated the process by loading the system with the pretrained BEATs' weights, which were unfrozen, enabling us to proceed with training the other components of the model from scratch. The complete pipeline is illustrated in Figure 3.

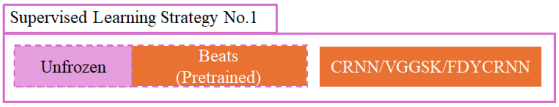

Fig. 3. Model Setting in Strategy 1

2) Strategy 2: Monolithic Framework with Initially Frozen, Then Unfrozen, Pre-trained Front-end: Initially, we configured the system with the BEATs model frozen and trained the other parts of the model from the ground up, a phase we designated as Stage 1. Subsequently, for Stage 2, we resumed from the Stage 1 model checkpoint, fine-tuning the entire model and unfreezing the BEATs component. Details of both stages are depicted in Figure 4.

E. Loss Function

The supervised loss is calculated by aggregating the binary cross-entropy (BCE) losses from two separate segments of the dataset. This BCE loss evaluates the divergence between predicted probabilities and true labels in both configurations of the model. For semi-supervised learning, Mean Square Error (MSE) is used.

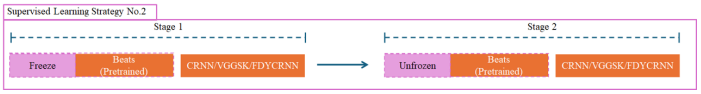

Fig. 4. Model Setting in Strategy 2

F. Adaptive Post-Processing

For sound event detection, we first apply binary thresholding $threshold = 0.5$ to the posteriors of the model output. Subsequent median filtering is used as post-processing to smooth the sequences. We adopt adaptive post-processing, using median filter window sizes W for each sound event class c. These sizes are calculated based on the actual durations of the events, defined as:

$$W_c = \text{duration} \cdot \beta \tag{1}$$

where the β was initially set to $1/7$.

III. Experiments

A. Dataset

The DESED [2] dataset consists of 10-second audio clips, either recorded in domestic settings or synthesized to emulate them, focusing on 10 sound event classes from AudioSet [12]. It includes a weakly labeled training set (1578 clips), an unlabeled in-domain set (10000 clips), and a synthetic strongly labeled set (10000 clips) as listed in Table I. Additionally, 3470 strongly annotated clips from AudioSet Strong share the same sound classes.

The MAESTRO [3][13] real dataset contains around 3-minute real-life recordings from five acoustic scenes, annotated via Amazon Mechanical Turk for soft label estimation as listed in Table. Training employs methods from the official baseline to align or adjust class labels across these diverse datasets. The MAESTRO [3][13] dataset features 17 sound categories, but training and evaluation focus on 11 chosen sound events due to variations in label confidence and quantity. These events include birds singing, car, people talking, footsteps, children's voices, wind blowing, brakes squeaking, large vehicle, cutlery and dishes, metro approaching, and metro departing.

TABLE I
DESED AND MAESTRO

DESED				
Subset	**Label Type**	**Sound Clips**	**Sampling Rate(kHZ)**	**Type**
Training Set	**Strong Label**	10,00	44.1	**Record**
	Strong Label	3,470	16	**Synthetic**
	Weak Label	1,578	44.1	**Record**
	Unlabeled	10,000	44.1	**Record**
Development	**Strong Label**	1,168	44.1	**Record**
Evaluation		699	44.1	**Record**
Maestro				
Training Set	**Strong Label**	7,503	16	**Record**
Development	**Strong Label**	3,474	16	**Record**

B. Evaluation Metric

Our system evaluation utilized the threshold-independent polyphonic sound event detection scores (PSDS), a primary metric since 2021 [14][15]. PSDS assesses the normalized partial area under the PSD-ROC curve, which averages class-specific ROC curves and includes a penalty for inter-class standard deviation. Key PSDS parameters include the detection tolerance criterion (ρDTC), the ground truth intersection criterion (ρGTC), the penalty weight (αST) on inter-class deviation, and the maximum false positive rate (emax). This year, we focused solely on PSDS1 for evaluation, setting DTC and ρGTC at 0.7, αST at 1, and emax at 100 FPs/hour, as PSDS2 is more suited for audio tagging.

For MAESTRO, we used segment-based labels with a segment length of one second and employed the segment-based mean partial area under the ROC curve (segMPAUC) as the primary metric, calculated with a maximum FP-rate of 0.1 and a binarization threshold of 0.5.

C. Baseline

As Table II illustrates, the discrepancy between the validation scores announced officially and those we obtained using the official baseline model. Moving forward, we will use these reconstructed baseline model results as the benchmark for comparison in subsequent sections.

TABLE II
PERFORMANCE COMPARISON OF CRNN+BEATS CONFIGURATIONS

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

D. Baseline Architecture with Diverse Parallel Front-End Feature Extractors

Table III presents a comparison of architectures featuring various front-ends. The results indicate negligible differences in PSDS1 scores; however, the FDYCRNN architecture exhibits a marginal improvement in mean pAUC. Accordingly, the BEATs model is subject to separate training and detailed analysis in subsequent sections.

TABLE III
PERFORMANCE COMPARISON OF CRNN+BEATS CONFIGURATIONS

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

E. Training Strategy with Mono Pre-trained Front-end

Table IV presents the performance of different training strategies as described in section D. The experimental results reveal that:

1) Directly fine-tuning the entire BEATs model for downstream tasks does not lead to optimal performance.:

2) Freezing the BEATs component initially and then training the rest of the model yields better results, particularly enhancing the PSDS1 performance.:

TABLE IV
RESULTS OF THE SINGLE MODEL WITHIN THE BEATS ARCHITECTURE ON THE DEVELOPMENT.

Model	PSDS1	mean pAUC
BEATs (Official)	0.47	0.72
BEATs-stage1 (Baseline)	0.49	0.73
BEATs-stage2 (Baseline)	0.50	0.73

3) Additional training in Stage 2 (BEATs unfrozen) shows limited further improvement.:

F. Experiment on Post-Processing

Tables V and VI present that applying a median filter for post-processing notably enhances the PSDS1 score, though its impact on mean pAUC is minimal.

TABLE V
PSDS1 OUTCOMES FOLLOWING THE COMPARISON OF POST-PROCESSED OUTPUTS FROM EACH MODEL.

Model	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 (Official)	0.20	0.47(+0.27)
BEATs-stage1 (Baseline)	0.23	0.49(+0.26)
BEATs-stage2 (Baseline)	0.39	0.50(+0.11)

TABLE VI
MEAN PAUC OUTCOMES WERE EVALUATED SUBSEQUENT TO THE COMPARISON OF POST-PROCESSED OUTPUTS FROM VARIOUS MODELS.

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)

IV. CONCLUSION

The performance outcomes of our systems on the development dataset as shown in Table VII. System 1 utilizes a two-stage training regimen with BEATs. Systems 2, 3, and 4 enhance category recognition by either averaging outcomes or selecting the top-performing category from candidate models. Specifically, System 2 employs unique ensemble methods for both DESED and MAESTRO categories. System 3 averages the results of the candidate models, while System 4 selects the most effective recognition category from these models. Systems 2 and 4 stand out, achieving the highest scores with a PSDS1 of 0.53 and a mean pAUC of 0.77. In further details, the final evaluation for a single model shows a PSDS of 0.495 and a mean pAUC of 0.733 on DESED, while ensemble models achieve a PSDS of 0.527 and a mean pAUC of 0.711 on MAESTRO.

TABLE VII
EVALUATION ON DEVELOPMENT

System	Ensemble	PSDS	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

V. CONCLUSIONS

The conclusion goes here.

REFERENCES

[1] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4 technical report," in *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2019.

[2] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *In Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[3] I. Mart´ın-Morato and A. Mesaros, "Strong labeling of sound events using crowd-sourced weak labels and annotator competence estimation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, 2023, pp. 902–914.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. L. Paz, "Mixup: Beyond empirical risk minimization," in *arXiv:1710.09412*, 2017.

[5] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, and B. Z. et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *arXiv:1904.08779*, 2019.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv:1409.1556.*, 2014.

[7] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 510–519.

[8] H. Nam, S. H. Kim, B. Y. Ko, and Y. Park, "Frequency dynamic convolution: Frequencyadaptive pat-tern recognition for sound event detection," in *arXiv:2203.15296v1*, 2022.

[9] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, and F. B. et.al, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *In Proceeding of Empirical Methods in Natural Language Processing*, 2019, pp. 1724–1734.

[10] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *In Proceeding of Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1195–1204.

[11] S. Chen, Y. Wu, C. Wang, S. Liu, Z. C. D. ompkins, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," in *arXiv:2212.09058*, 2018.

[12] A. Tarvainen and H. Valpola, "Audio set: An ontology and human-labeled dataset for audio events," in *In Proceeding of IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 1195–1204.

[13] I. Mart´ın-Morato, M. Harju, P. Ahokas, and A. Mesaros, "Training sound event detection with soft labels from crowdsourced annotations," in *In Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[14] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *In Proceeding of IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 61–65.

[15] J. Ebbers, R. Serizel, and R. H. Umbach, "Threshold independent evaluation of sound event detection scores," in *In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 1021–1025.