SOUND EVENT DETECTION BY AGGREGATING PRE-TRAINED EMBEDDINGS FROM DIFFERENT LAYERS

Technical Report

Xuenan Xu, Ziyang Ma, Fei Yang, Guanrou Yang, Mengyue Wu, Xie Chen

MoE Key Lab of Artificial Intelligence
X-LANCE Lab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
{*wsntxxn, zym.22, yangguanrou, mengyuewu,chenxie95*}@sjtu.edu.cn, fyang20528@gmail.com

ABSTRACT

This technical report is the system description of the X-Lance team submission to the DCASE 2023 task 4b challenge: sound event detection with soft labels. Our submissions focus on incorporating informative audio representations from self-supervised learning. The embeddings from different layers of the pre-trained models are aggregated as the input of our model. Since the occurrence of sound events in different scenes is imbalanced, for each scene we train our models using all the audio files. Finally, models of different architectures trained under different scenes are ensembled with learned weights.

***Index Terms*—** Sound event detection, Pre-training model, Self-supervised learning, Convolutional recurrent neural networks

1. INTRODUCTION

This paper proposes our system for the DCASE 2023 task4b challenge, which is concerned with training sound event detection (SED) systems with soft labels. The goal of SED is to automatically identify and localize specific sound events within an audio recording [1]. Popular SED architectures include convolutional neural networks (CNN) [2], convolutional recurrent neural networks (CRNN) [3] and Transformers [4].

Soft labels refer to a type of annotation in which a sound event is labeled with probabilities of the set of all possible sound event classes, rather than with a hard label indicating a single class. Soft labels are used in sound event detection to indicate the certainty of human annotators. They may be useful to train SED systems since there is uncertainty or ambiguity in the classification of sound events, especially for background sounds. In previous versions of the DCASE task 4 challenge, the development dataset is composed of either weakly labeled or unlabeled real data and strongly labeled simulated data. This subtask aims to explore the impact of introducing soft labels in the training process. However, the dataset size is small and we find that the dataset is highly imbalanced regarding sound events. The distribution in different scenes (see Section 3) is uneven. For example, "metro approaching" and "metro leaving" only appears in the scene "metro station". To enable more samples to be used in training, we train different models under different scenes. For each scene, all data is used for training so even the most infrequent classes contain a moderate number of training samples. Moreover, the occurrence numbers of different events are also imbalanced: there are 5703 "people talking" segments annotated with a prob of over 0.5 while the number for "children voices" is only 183. Therefore, we mainly explore transferring the knowledge learned by self-supervised learning on large-scale datasets to this task. We train two models based on features extracted from pre-trained models and ensemble them to make final predictions.

The report is structured as follows. Section 2 describes our core system design including feature engineering and model architectures. Next, Section 3 introduces the experimental setup and our submission details. Finally, Section 4 concludes our work.

2. SYSTEM

2.1. Pre-trained Audio Representations

In this challenge, we incorporate BEATs [5] to extract informative representations from the input audio. The BEATs model utilizes a self-supervised learning paradigm to predict discrete units generated by an acoustic tokenizer, which presents superior performance on various audio-related downstream tasks. Since the BEATs model is not included in the external model resources, we re-implement the BEATs model and train it on AudioSet [6]. Next, we use the BEATs model to generate the audio features from the original waveform given by this challenge. In this way, we can take advantage of a large amount of unlabeled audio data and obtain general audio features. BEATs first converts an audio clip into the mel-spectrogram with a stride of 10ms and a dimension of 128, and then divides it into patch-level embedding sequences. The output of the BEATs model is 768-dimensional features with a stride of 20ms. We consider two types of pre-trained features:

Clip-level Pre-trained Features. Average pooling is used on the whole sequence to obtain the clip-level pre-trained feature. We concatenate this global feature with each frame-level FBank feature and feed them into subsequent networks. The features are used in the CRNN architecture mentioned in Section 2.2, and for Submission 2 and Submission 4 mentioned in Section 3.4.

Frame-Level Pre-trained Features. The features generated by the BEATs model with a stride of 20ms are directly used for the LSTM architecture mentioned in Section 2.2, and for Submission 3 and Submission 4 mentioned in Section 3.4.

Since BEATs is a multi-layer transformer architecture, all the above features come from the weighted average of the multi-layer transformer outputs.

2.2. SED Model

Two types of architectures are used in our systems.

CRNN architecture. The models of the CRNN architecture are a slight modification on the challenge baseline. The baseline model is a CRNN with a 3-layer CNN and a single-layer bidirectional gated recurrent unit (GRU) network. Clip-level BEATs embeddings from each layer are aggregated to obtain a single clip-level embedding. The aggregation weights are learned during training. Then the embedding is concatenated with the feature after convolution blocks. Finally, the GRU predicts the probabilities of each event.

LSTM architecture. The models of the LSTM architecture directly use frame-level BEATs embeddings as the only input. Similarly, embeddings from different layers are aggregated using learned weights. Then a single-layer bidirectional long short-term memory (LSTM) network is used.

2.3. Ensembling

We use the stacking strategy to ensemble the models. We concatenate the probabilities output by different models, and then train a fully connected layer to predict the ground truth soft label. We use the mean of different model probabilities to initialize the fully connected layers and set the learning rate to a small number. These ensure that the stacking model will not learn trivial weights.

3. EXPERIMENTAL SETUP

3.1. Dataset

In this challenge, MAESTRO real [7] is used as the development set. There are 49 audio recordings in total, with a duration of 3.16 hours. They are recorded in 5 different scenes: cafe restaurant, city center, grocery store, metro station and residential area. Multiple annotators are instructed to estimate soft labels with a time resolution of 1s. Since the dataset size is very small, 5-fold cross validation is used in the baseline setting. To use more data for training, we use both the original training and validation sets as the training set and the original test set as the validation set. After training, models from all folds are ensembled.

3.2. Training Hyper-parameters

For the RCNN model, we use the same setting as the baseline. 64-dimensional log mel-spectrogram with a window size of 0.4s and a window shift of 0.2s is extracted from the audio as the input feature. For the LSTM model, the 768-dimensional BEATs with a window shift of 0.02s features are fed into the network with a 128-dimensional hidden layer. For both models, original audio recordings are split into segments of 40s during training and the batch size is set to 32. We use average pooling for downsampling and the output time resolution is 1s. Models are trained with a learning rate of 0.001 for at most 100 epochs and an early stop patience of 10 using the Adam optimizer. The learning rate will be set to its $\frac{1}{10}$ if the validation loss does not decrease for 5 epochs.

3.3. Evaluation Metrics

The challenge uses macro-average segment-F1 ($F1_{MO}$) score [8] under the optimum threshold as evaluation metrics. Segment-level F1 evaluates the accuracy of model predictions in each 1s segment. A segment will be considered active for an event if any frame in this segment receives a high score. Therefore, our models use a large time resolution (0.2s/1s) to reduce false alarms. The optimum threshold will be searched for each class on the evaluation data using the toolkit [9] so the magnitude of estimated probabilities is not necessarily close to 1 or 0.

3.4. Submissions

Here are details of our submissions:

Submission 1: The submission without using external data. We use a similar setting as the baseline. It achieves an $F1_{MO}$ of 55.79.

Submission 2: The CRNN model architecture. Models trained on the 5 folds are averaged. It achieves an $F1_{MO}$ of 59.88.

Submission 3: The LSTM model architecture. Models trained on the 5 folds are averaged. It achieves an $F1_{MO}$ of 57.25.

Submission 4: Different models are trained for each scene. In each scene, all files are split into several folds. Finally, 47 models in total are ensembled using learned weights with tricks mentioned in 2.3. It achieves an $F1_{MO}$ of 69.85.

It should be noted that since we use all data for training and validation, our reported results are not comparable to results under the official development-test setup.

4. CONCLUSION

This paper summarizes our submission to the DCASE2023 task4b challenge. Our approach is based on pre-trained BEATs features. The pre-trained features are incorporated with the model in either clip-level or frame-level ways. To use more training data for sound events with few samples, we train separate models under different scenes. Finally, all models are ensembled by learnable weights.

5. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized decision surface and disentangled feature for weakly-supervised polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1466–1478, 2020.

[3] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[4] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[5] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[6] 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 *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[7] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

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