FEW-SHOT EMBEDDING LEARNING AND EVENT FILTERING FOR BIOACOUSTIC EVENT DETECTION

Technical Report

Jigang Tang[1]*, Xueyang Zhang*[1]*, Tian Gao*[1]*, Diyuan Liu*[1]*, Xin Fang*[1]*, Jia Pan*[1]*,*
Qing Wang[2]*, Jun Du*[2]*, Kele Xu*[3]*, Qinghua Pan*[1]

[1] iFLYTEK Research Institute, Hefei, China, {tiangao5, dyliu2}@iflytek.com
[2] University of Science and Technology of China, HeFei, China, {qingwang2, jundu}@ustc.edu.cn
[3] National University of Defense Technology, xukelele@163.com

ABSTRACT

In this technical report, we describe our submission system for DCASE2022 Task5: few-shot bioacoustic event detection. We propose several methods to improve the representational ability of embedding under limited positive samples. Including the segment-level and frame-level embedding learning strategy, model adaptation technology and embedding-guided event filtering approach. The event filtering task is independently trained on each test file to improve the discrimination of embeddings between similar events. The proposed system is evaluated on the official validation set, and the best overall F-measure score is 74.4%.

***Index Terms*—** DCASE, few-shot bioacoustic event detection, embedding learning, model adaptation, event filtering

1. INTRODUCTION

Few-shot learning (FSL), expecially few-shot image classification (FSIC) has received considerable attention [1]. The existing typical FSIC methods can be roughly divided into three categories, including semi-supervised based [2], transductive based [3, 4] and inductive based [5]. The few-shot bioacoustic event detection essentially can be seen as a FSIC task, where lots of query sets for each audio are retrievable. Thence, we develop several solutions from the perspective of semi-supervised supervision. Based on pretraining-finetuning process scheme, three mothods are proposed in this work. First, the segment-level embedding learning stratery. Second, the frame-level embedding learning method. And the third, embedding-guided event filtering approch.

To get a "negative" category center (also called "negative" prototype), the baseline system assumes that the density of positive events is low, so the whole audio is selected as the negative set. In addition, the "negative" prototypes are generated by random sampling. However, we found this hypothesis has low reliability in some conditions. For some audio with a large proportion of positive samples, some positive segments will be regarded as negative samples due to the negative samples are randomly selected, which will lead to poor test results. Therefore, an appropriate adjustment is made in this work. We assume the time period in the middle of five labeled supports and before the first labeled support have higher reliability to be selected as negative samples.

Duration of suport segments varies from a few milliseconds to a dozen seconds in the validation set. The 2022 baseline system provides an adaptive window length and adaptive window shift scheme.

Table 1: Adaptive window length and fixed window shift scheme.

X	$window\ length$	$window\ shift$
X $\leq$ 17	17	4
17 $<$ X $\leq$ 100	X	4
100 $<$ X $\leq$ 200	X // 2	4
200 $<$ X $\leq$ 400	X // 4	4
X $>$ 400	X // 8	4

However, when the window shift is set to half the window length, the number of audio segments to be retrieved is much smaller, that is, the discrimination resolution of long audio is too low. In our segment-level methods of schemes 1 and 3, an adaptive window length and fixed window shift (ALFS) is used. The setting method is shown in Table 1, where X represents the mean of 5 positive support durations, $X = mean(dura(support))$.

As shown in table 1, the shortest window length is set to 17 frames, which is 0.2 seconds. However, most of the audio segments in the support set are only 0.02 to 0.05 seconds in PB class. And it is difficult for the segment-level scheme to extract credible representative embedding. Thence, we propose the frame-level method, that is our scheme 2. Retrieval resolution is enhanced by determining whether target event exist in each frame.

2. FRAMEWORK OF DETECTION

Our detection process is roughly as shown in Figure 1. According to the positive label and the specific negative segments selection method, each audio in the test set is divided into positive segments, negative segments, and query sets. The PCEN features are performed on the spectrograms of audio segments. Hereafter, the above PCEN are input into embedding extraction network to obtain the segment-level (or frame-level for frame-level method) embedding representation. Then the prototype features are obtained by taking embdding mean of "postive" and "negative". The positive and negative central embedding are spliced to form a 1024*2 feature vector, which is used as an initialization parameter of the softmax binary classifier, that is $W \in R^{2\times d}$. Finally, the embedding of query set is multiplied by W, and the prediction results is obtained through a softmax function. Finally, the F1 results of our three systems are 74.4%, 68.2%, and 58.2%, respectively. Detail results are shown in Table 2.

Table 2: Detailed validation results including four classes.

System	*Precision*(%)	*Recall*(%)	*F1*(%)	$F1-HB(\%)$	$F1-ME(\%)$	$F1-ML(\%)$	$F1-PB(\%)$
Baseline	36.3	24.9	29.5	/	/	/	/
Frame-Level	77.5	71.5	74.4	77.0	90.0	90.5	53.7
Seg-Level	75.6	62.1	68.2	85.8	79.2	73.5	48.1
Event-Filter	66.5	51.8	58.2	76.7	64.2	72.9	42.4

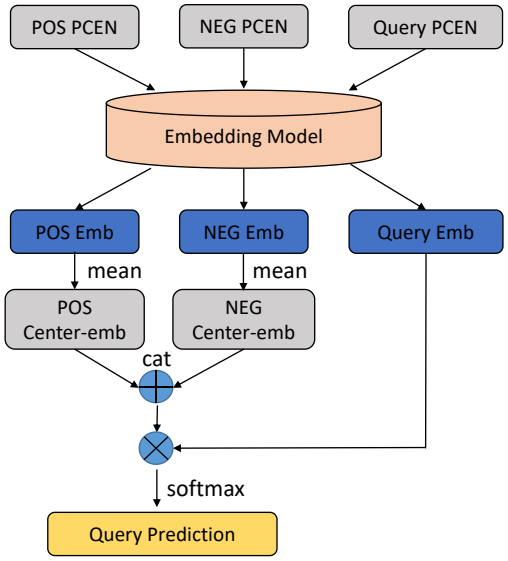

Figure 1: The system framework of few-shot bioacoustic event detection.

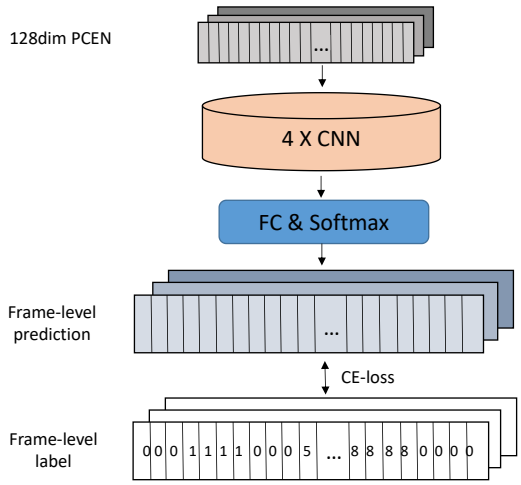

Figure 2: The framework of frame level model.

Table 3: The network architecture of frame-level embedding model.

Block	*kernel_stride*	*BN_Activate*
CNN_Block1	conv,3 × 3, (1, 128)	BN+ReLU
CNN_Block2	conv,3 × 3, (128, 128)	BN+ReLU
CNN_Block3	conv,3 × 3, (128, 128)	BN+ReLU
CNN_Block4	conv,3 × 3, (128, 128)	BN+ReLU
FC	fc(1024, 2)	Softmax

3. FRAME-LEVEL EMBEDDING LEARNING

The training and testing framework of frame-level embedding learning system are respectively shown in Figure 2 and Figure 3. Compared to the baseline segment-level system [1], the frame-level method can make use of the similar relations among the adjacent frames. Due to the stable feature embedding can be extracted of variable length audio, we found that the frame level framework can obtain a better adaptive system during fine-tuning. In addition, a two-step fine-tuning scheme is designed in the testing stage, which can make use of train-test set and get better feature embedding.

The frame level system. The network structure is shown in Figure 2, in detail, which consists of 4 CNN layers and 1 linear layer. Different from the baseline system, a larger channel number (128 channels) is used.Meanwhile, to get a prediction of each frame feature, the maxpooling downsampling layer is removed. The specific configuration is shown in Table 3. We preform Per-Channel Energy Normalisation (PCEN) on the Mel spectrograms of 128 bins [6], with the 1024 FFT samples and a hop of 256 samples. In the training stage, we use simples CE loss rather than few-shot loss functions. Since the 2022 training set cannot be converged during training, we removed the WMW class and the left audios contain 19 kinds of animal calls, which is used as our training set.

Two-step adaptive strategy. As shown in Figure 3, there are two-steps during fine-tuning. First, the 5-shot labeled support segments are selected as positive set, and the four segments between two positive samples are selected as negative set. We use cross entropy loss function to distinguish the two classes. In order to obtain better feature representation, we combine training set and the positive samples to define a 20-classification task.The second-step, we can get posterior probability of query set by the first-step model. Through method of fixed threshold selection, we set a high threshold to filter query results with high confidence into the positive set, thereby increasing the number of positive examples for training. Then repeat step1 and step2 until a set number of iterations. With the frame-level embedding learning, we got a powerful result in the development set as table 2, which F1-score is 74.4%.

4. SEGMENT-LEVEL EMBEDDING LEARNING

Our pre-training fine-tuning segment-level scheme is shown in Figure 4. The overall network consists of 4 ConvBlocks ProtoNet [7],

[1] https://github.com/c4dm/dcase-few-shot-bioacoustic/tree/main/baselines/deep_learning

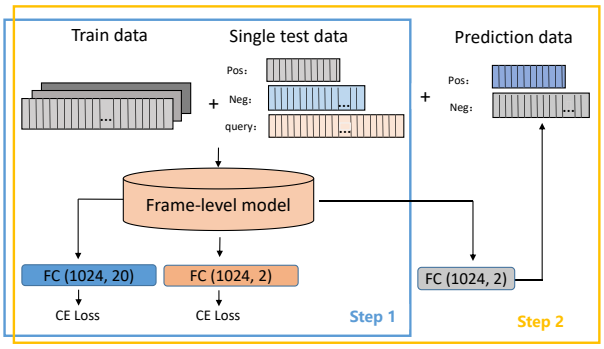

Figure 3: The framework of semi-supervised learning.

an adaptive average pooling and a fully connected layer. Each ConvBlocks is made up of a conv2d, a Batch Normalization and a ReLU activation layer. During training, the 128 dimensional PCEN features are divided into 0.2 seconds (17 frames) per segment, and the loss function is CE Loss.

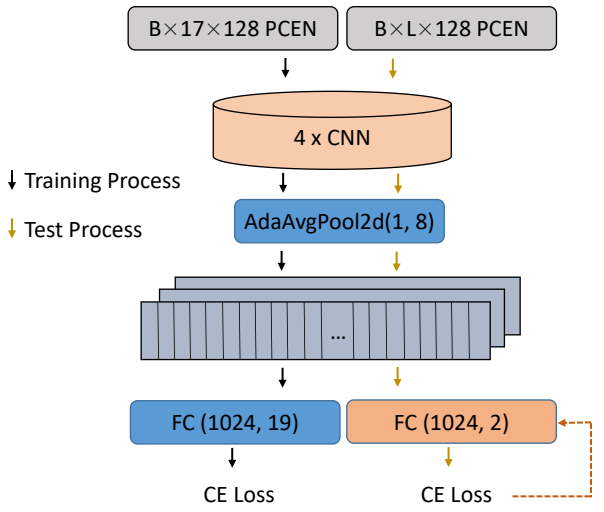

Figure 4: The framework of Adaptive-Length Segment-level model.

In Figure 4, the black arrows represent the training process, and the gold arrows represent the testing or fine-tuning process. When the training is completed, backbone parameters are fixed, the FC(1024, 2) layer is initialized by the "postive" and "negative" prototype features. Fine-tuning process of FC(1024,2) sets the number of iterations to 5, uses the Adam optimizer with learning rate equal to 1e-5. Different iterations or learning rates have different performances on different validation audio, but we did not perform targeted debugging on per validation category.

Since the adaptive window length and fixed window shift scheme are used, conversion method from onset (offset) frame number to time needs to be modified appropriately. The onset times of the retrieval results of the baseline system are calculated as Equation 1, and our revised calculation method is as shown in Equation 2. Where $onset\_f$ means the onset frame, hop means window shift (hop=4 is this work), seg mean adaptive window length.

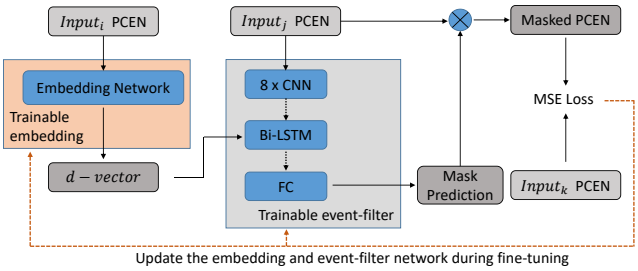

Figure 5: The framework of event filtering model.

Table 4: The data configuration of event filter model.

Embedding	Input	Target
Positive	Positive	Positive
Positive	Positive + Negative	Positive
Positive	Positive + Query	Positive
Positive	Negative	zero

$$\begin{aligned} onset &= (onset\_f + 1) * const \\ offset &= (offset\_f + 1) * const \end{aligned} \tag{1}$$

$$\begin{aligned} onset &= (\lfloor seg * 0.66 \rfloor + (onset\_f - 1)) * const \\ offset &= (\lfloor seg * 0.66 \rfloor + (offset\_f - 1)) * const \end{aligned} \tag{2}$$

$$const = hop * 256/22050 \tag{3}$$

5. EVENT FILTER AND MODEL ADAPTATION

Our event filter model is shown in Figure 5. The overall framework is similar with VoiceFilter[8], which contains an embedding extraction module and a separation module. The embedding extraction module is the same as segment-level embedding learning scheme. The separation module consists of CNN, BLSTM and FC layers. For specific configuration parameters, please refer to the literature[8]. Given the embedding input, the event filer model are expected to extract postive samples from the input mixture data. Tabel 4 lists the data configuration of event filter model training. In the testing stage, the event filter task is independently trained on each test file to improve the discrimination of embeddings between similar events. As shown by the red dotted line in the Figure 5, during the event filter training process, the embedding extraction network will update the parameters synchronously. We treat this process as model adaptive training on unseen classes. In the final detection stage, the original and filtered data are fed into the embedding network to generate embeddings. Embeddings are then averaged. Subsequent processing is the same as above.

6. REFERENCES

[1] W. Li, C. Dong, P. Tian, T. Qin, X. Yang, Z. Wang, J. Huo, Y. Shi, L. Wang, Y. Gao, *et al.*, "Libfewshot: A comprehensive library for few-shot learning," *arXiv preprint arXiv:2109.04898*, 2021.

[2] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, "Instance credibility inference for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 836–12 845.

[3] E. Lee, C.-H. Huang, and C.-Y. Lee, "Few-shot and continual learning with attentive independent mechanisms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9455–9464.

[4] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," in *European Conference on Computer Vision*. Springer, 2020, pp. 121–138.

[5] Z. Zhou, X. Qiu, J. Xie, J. Wu, and C. Zhang, "Binocular mutual learning for improving few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8402–8411.

[6] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, "Robust sound event detection in bioacoustic sensor networks," *PloS one*, vol. 14, no. 10, p. e0214168, 2019.

[7] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," 2018.