2023 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-6891-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICME55011.2023.00134

Frame-Level Embedding Learning for Few-shot Bioacoustic Event Detection

1st Xueyang Zhang
iFlytek Research
Hefei, China

2nd Shuxian Wang
University of Science and Technology of China
Hefei, China

3rd Jun Du*
University of Science and Technology of China
Hefei, China

4th Genwei Yan
China University of Mining and Technology
Xuzhou, China

5th Jigang Tang
iFlytek Research
Hefei, China

6th Tian Gao
iFlytek Research
Hefei, China

7th Xin Fang
iFlytek Research
Hefei, China

8th Jia Pan
iFlytek Research
Hefei, China

9th Jianqing Gao
iFlytek Research
Hefei, China

***Abstract*—We propose an effective frame-level embedding learning framework for few-shot bioacoustic event detection (FSBED). First, the duration of different animal calls varies greatly, so we innovatively propose a frame-level embedding learning scheme, which can obtain adaptive event receptive fields with more accurate frame-level units. Next, we develop a transfer learning-based approach to deal with the mismatch between training and testing data. Finally, we use the idea of semi-supervised learning to solve the problem of too little labeled data in few-shot learning. By incorporating these several sets of techniques, our overall system ranked first place in the FSBED task of Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2022.**

***Index Terms*—DCASE, few-shot bioacoustic event detection, frame-level embedding learning, transfer learning**

I. Introduction

Few-shot bioacoustic event detection (FSBED) is a task that focuses on sound event detection (SED) in few-shot learning (FSL) setting for the animal (mammal and bird) vocalisations [1], [2]. Obviously, the vocalisations of different animals are diverse, and audio annotation is costly and time-consuming. Therefore, compared with general SED [3]–[9] tasks, it is difficult to obtain a large amount of labeled data in bioacoustic event detection (BED) tasks. Thus, FSL [10]–[16] is introduced into the BED task. FSL describes tasks in which an algorithm must make predictions given only a few instances of each class. Specifically, FSL is usually studied using *N-way-k-shot* classification, where N denotes the number of classes and k denotes the number of known examples for each class. For the FSBED task, we need to extract information from five exemplars (i.e., $k = 5$) vocalisations (shots) of mammals or birds and detect and classify sounds in field recordings [1], [2]. It has a wide range of applications, for example, it can greatly save the time of biologists annotating very long audio recordings, and then they can monitor biodiversity and animal behavior based on the annotated audio recordings.

In recent years, the FSBED task has attracted a lot of attention. The mainstream methods for this task are as follows. One approach is template matching [17], which is commonly used in bioacoustics based on spectrogram cross-correlation. This approach performs event detection based on the normalized cross-correlation between labeled sound events and unlabeled audio recordings [1], [2]. Another approach is a prototypical network [11] that aims to learn a classifier that can quickly adapt to new classes with only a few examples. This method has been widely used in FSBED tasks [1], [2], [18]–[24]. Although the prototypical network has achieved certain results in this task, there are still some deficiencies that can be improved. Firstly, since the support set has only a small number of labeled samples, the class prototypes may not accurately represent the class centers. Secondly, the feature extractor is task-agnostic (or class-agnostic): the feature extractor is trained with base-class data and directly applied to unseen-class data [25]. Therefore, Yang [25] et al. proposed a mutual learning framework with transfer learning aimed at iteratively updating class prototypes and feature extractors. In addition, some teams explored the role of data augmentation on FSBED tasks [26], [27].

Although the above methods have achieved some success on the FSBED task, they usually adopt a segment-level approach to detect the presence or absence of a target sound event. However, the official dataset contains recordings of vocalisations from different species of animals [1], [2], and their vocalisation durations vary widely. For example, most audio segments in the support set are only 0.02 to 0.05 seconds in the Polish Baltic Sea bird flight calls (PB) class of the validation set [2]. It is difficult for segment-level schemes to extract

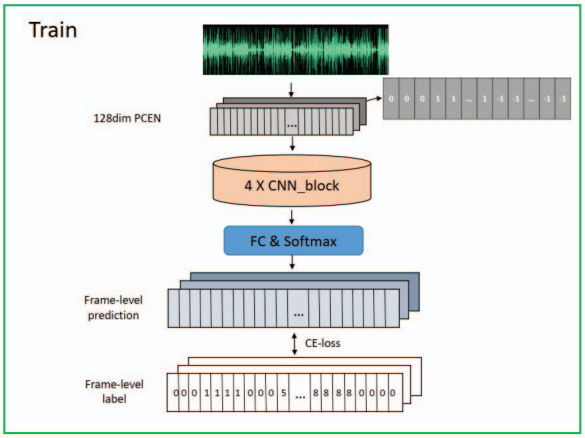
*corresponding author

1945-788X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICME55011.2023.00134

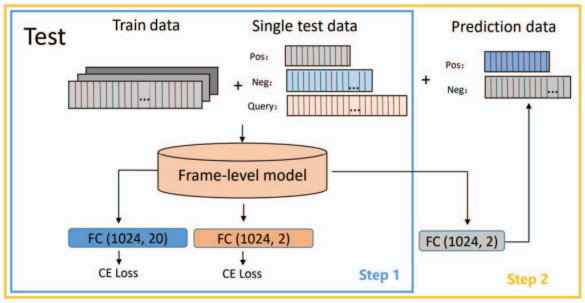
credible representative embeddings. Therefore, we propose a frame-level framework. In addition, this task also suffers from the training-test set mismatch and too little labeled data. Therefore, we introduce some ideas of transfer learning and semi-supervised learning into this task. Our contributions can be summarized as follows:

(1) We propose a frame-level embedding learning framework to improve retrieval resolution.
(2) Based on transfer learning, we use few labeled segments for joint training with the training set to deal with data mismatch.
(3) We introduce the idea of semi-supervised learning, and add some high-confidence data predicted by the model after one-stage fine-tuning as positive (POS) samples to the next stage of training to reduce the impact of too few labeled samples.

II. Method



(a) The training framework



(b) The testing framework

Fig. 1. The framework for training (a) and testing (b) of frame-level embedding learning method. Step1 in (b) represents based on transfer learning, we use few labeled segments for joint training with the training set. Step2 indicates that semi-supervised learning is introduced, and some high confidence data predicted by the model after one-stage fine-tuning is added to the next stage of training as positive samples. The upper right corner of (a) is a sliding window randomly selected. The two FC (1024, 2) in (b) are the same classifier.

The overall framework of our proposed FSBED method is shown in Fig. 1. First, we use labeled data to train a frame-level backbone model, as shown in Fig 1(a). Next, in the test stage, since the correlation between the training set and the test set is extremely low, we use few labeled fragments of the test data to perform joint training with the training set, as shown in Step1 in Fig. 1(b). FC (1024, 2) is the classification weight of positive and negative examples in the test audio, and FC (1024, 20) is a 20-class classifier composed of 19 classes in the training set and positive examples in the test audio. The classification loss generated by the two classifiers is jointly backpropagated to update the model. Finally, we select some high-confidence data as positive examples from the prediction results generated by the model after fine-tuning in the first stage and add them to the next stage of training to reduce the impact of too few labeled samples, as shown in Step 2 in Fig. 1(b).

A. Frame-Level Learning Framework

Existing methods for the FSBED task often use segment-level modeling, which lacks the ability to capture short-term events. We analyze this because it predicts at the segment level in units of 17 frames, so it cannot accurately predict some extremely short events (5 frames). Therefore, we propose the frame-level system.

Our model uses 4 CNN layers of [11], and in order to obtain finer granularity on the time axis, we remove the last two MaxPool and add the repeat-interleave operation. The model is shown in Table I.

TABLE I
THE NETWORK ARCHITECTURE OF FRAME-LEVEL EMBEDDING LEARNING MODEL.

Block	Kernel	Channel	Activation
CNN_block1	conv, 3×3	(1, 128)	BN+ReLU
MaxPool2d(2)	-	-	
CNN_block2	conv, 3×3	(128, 128)	BN+ReLU
MaxPool2d(2)	-	-	
CNN_block3	conv, 3×3	(128, 128)	BN+ReLU
CNN_block4	conv, 3×3	(128, 128)	BN+ReLU
FC	FC(1024, 2)		Softmax

In the training stage, we first perform Per-Channel Energy Normalisation (PCEN) [28] with a sampling rate of 22050Hz. 431 window length and 86 window shifts (11ms for each frame) are adopted for sliding window segmentation of PCEN to get a large receptive field, and each frame in the window is labeled, then we feed it into our frame-level model. After repeating the representations 4 times to recover the length on the time axis at the end of the model, the Cross-Entropy (CE) loss is calculated as follows:

$$L = -\frac{1}{k \times B} \sum_{i=1}^{b} \sum_{j=1}^{c} (y_{i,j} \odot \mathbb{M}_i) \log(f(x_i)) \qquad (1)$$

where $y_i = \left(y_i^1, y_i^2, ..., y_i^k\right)$, $y_i^k \in \{0, ..., 19\}$ is the ground truth of one frame, and k is the window length. Specifically, "1∼19" represents the label of 19 types of bioacoustic events in the training set, and "0" represents background events.

$\mathbb{M}_i \in \{0, 1\}$ denotes the training mask for frames in a window, and "0" is for the overlapped frame of the POS event between adjacent windows. $x_i = \left(x_i^1, x_i^2, ..., x_i^k\right)$, x_i^k is one frame (128-dimensional PCEN features). $\odot$ denotes the dot product by frame, f is the frame-level embedding network. b is the total number of windows, and B is the total number of valid windows in which $\mathbb{M}_i$ is not all 0. j represents the class index, c is the total number of classes (i.e., 20).

In the test stage, we segment the data before the end of the 5th shot in support. X_p=$\left\{\left(x_i, y_p^k\right)\right\}_{i=1}^{N_p}$, where x_i denotes a POS segment, y_p^k is the label of one frame in x_i, which is 1. N_p is 5. We select the segments between X_p as negative (NEG) set, X_n=$\left\{\left(x_j, y_n^k\right)\right\}_{j=1}^{N_n}$, where x_j denotes a NEG segment, y_n^k is the label of one frame in x_j, which is 0. N_n is 5. $X_p \cap X_n = \emptyset$. $X_s = X_p + X_n$, $X_q = X - X_s$, where X, X_s and X_q represents the whole audio, support set and query set, respectively. We create a new 2-classifier to define a 2-classification on the X_s, and fine-tune the 2-classifier until a set number of iterations.

B. Weight Sharing Transfer Learning

Considering that there may be differences between the source domain (training set) and the target domain (test set), fine-tuning the network only on the target domain may skew the feature extractor. Therefore we combine the training set and the POS in the test set to define a 20-classification task. $X_{\mathrm{TL}} = X_t + X_p$, where X_t denotes the training set. Inspired by [29], we share the weight of the 0 class in the 20-classifier and the weight of the POS class in the 2-classifier. $W_p = \left(w_{p0}^T, w_{p1}^T\right)$, $W_t = \left(w_{t0}^T, w_{t1}^T, ..., w_{t19}^T\right)$, where the W_p is the weight of 2-classifier, w_{p1}^T is the weight of the POS class in the 2-classifier, W_t is the weight of the decoder which is a 20-classifier, w_{t0}^T is the weight of the POS class of the test set in the 20-classifier. The w_{p1}^T shares the same tensor with the w_{t0}^T. In order to strengthen the classification ability of POS class, we fine-tune the 20-classifier and the 2-classifier together.

C. Semi-Supervised Learning

As mentioned in [25], incomplete support sets data will lead to the network cannot represent the category center, and fine-tuning the model only in support is easy to overfitting. One way to solve this problem is finding supplementary information, which can help the representation of model as close to the true category center as possible. Inspired by [30], we propose a pseudo-label filtering method with an adaptive threshold. Specifically, we first calculate the ratio of POS predicted correctly in each window in X_s to the total POS in the window, and take the minimum ratio as the threshold of the pseudo-label. Then, the softmax value of each frame in X_q is screened by threshold-value ($thre$), and the pseudo-label value is set to 1 if it is larger than the $thre$, or 0 if it is smaller than the $thre$-0.2, otherwise -1, as shown in Eq. 2, where p is the softmax output of the POS class in the 2-classifier, and v is 0.2. The calculation method of the $thre$ is shown in Eq. 3, where j is the window index, N_s is the total number of windows in X_s, n_j is the length of the j-th window, and y_i is the label of the i-th frame x_i in a window (the labels of POS and NEG frames are 1 and 0, respectively).

After the frames in the query set X_q are pseudo-labeled, the frames marked as 1 (POS) and 0 (NEG) are added to the next stage of training together with the support set X_s and training set X_t, while the frames marked as -1 in X_q are not used for training. Finally, the total loss function is shown in Eq. 4, where L_s is the CE loss of X_s, L_q is the CE loss of X_q, and L_t is the CE loss of X_{TL}, which is the same as Eq. 1.

$$y = \begin{cases} 1 & if \quad p > thre \\ 0 & if \quad p < thre - v \\ -1 & others \end{cases} \tag{2}$$

$$thre = \min_{j=1,...,N_s} \frac{\sum_{i=1}^{n_j} y_i P(\hat{y} = 1 \mid x_i)}{\sum_{i=1}^{n_j} y_i} \tag{3}$$

$$L_{\mathrm{total}} = L_s + L_q + L_t \tag{4}$$

III. Experiments and Analysis

A. Dataset Analysis

The dataset we use comes from DCASE 2022 Task5 [2], which contains development set and evaluation set, and the development set is divided into the training set and validation set. They are derived from the sounds of many different animals, including birds, hyenas, meerkats, jackdaws, humbug, and more. The training set contains multi-class temporal annotations, provided for each recording: positive (POS), negative (NEG) and unknown (UNK). For the validation set, only one-class temporal annotations (POS/UNK) were provided for each record. During the challenge, the evaluation set provided only the top five POS events of the category of interest for each recording. Our results on the evaluation set are available on the DCASE 2022 Task5 Challenge results page[1]. The full labels of the evaluation set have not been released during the challenge, so in this paper, we evaluate our method on the validation set. The evaluation of this task is based on an event-level F-measure with macro-averaged metric across all classes [2].

In the DCASE 2021 challenge, there is also the FSBED task[2]. By analyzing this task in 2021 and 2022, it can be found that the evaluation indicators for the two years are consistent, but the datasets in 2022 have become more difficult. The official validation set and evaluation set details are shown in Table II. The sub-folders in Table II represent different scenarios in the validation set and evaluation set, and different scenarios correspond to different collections of bioacoustic sources. The validation sets for 2021 and 2022 contain 2 (HV, PB) and 3 (HB, PB, ME) different scenarios, respectively, while the evaluation sets for 2021 and 2022 contain 3 (DC,

[1]https://dcase.community/challenge2022/task-few-shot-bioacoustic-event-detection-results

[2]https://dcase.community/challenge2021/task-few-shot-bioacoustic-event-detection

ME, ML) and 6 (DC, CT, CHE, MGE, MS, QU) different scenarios, respectively [1], [2]. As can be seen from Table II, the difficulties of the DCASE 2022 datasets are mainly reflected in: 1) scenarios are more varied and complex; 2) the number of events to be detected increases.

TABLE II
VALIDATION AND EVALUATION SET STATISTICS FOR THE FEW-SHOT BIOACOUSTIC SOUND EVENT DETECTION TASKS OF DCASE 2021 AND DCASE 2022.

Dataset	Validation		Evaluation	
Years	2021	2022	2021	2022
Total Audio Recordings	8	18	31	46
Total Classes	4	5	22	13
Total Sub-folders	2	3	3	6
Total Events	310	1077	2072	9605

B. Ablation Study

1) Experiments on Segment-Level Framework: **Baseline System.** We adopt the transductive inference-mutual learning (TI-ML) framework of the DCASE 2021 FSBED first-place team as our baseline system [25]. We retrain it on the DCASE 2022 dataset and evaluate it on the 2022 validation set. The experimental settings here are the same as [25]. In order to improve the diversity of data and the generalization of the model, we employ the mixup [31] data augmentation in the training stage. The experimental results are shown in Exp No.1 and 2 of Table III, respectively. The total F-score in Exp No.2 increased by 8.3%.

The use of mixup enables our network to generalize well enough, but in the ML framework, heavy fine-tuning of networks with insufficient support may cause overfitting problems. Therefore, we remove the ML process and only slightly fine-tune the 2-classifier. The results are shown in Exp No.3 of Table III. Compared with Exp No.2, the F-score is increased by 13.2%, which shows that our method is effective.

Negative Sample Selection. To get a "negative" category center (also called "negative" prototype), the baseline system assumes that the density of POS events is low, so the whole audio is selected as a 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 POS samples, some POS 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. The result is shown in Table III Exp No.4. Compared with the result in Exp No.3, the total F-score increased by 1.3%. In addition, after this strategy is used, we find that the stability of our model output is greatly improved compared with before.

Sliding Window Strategy Adjustment. Through the visual analysis of POS segments marked of each audio, we find that the length of POS segments in different audio differs greatly, but in the same audio is basically the same. It is hard to set a suitable fixed window length for all audios, so we set an adaptive window length according to the length of POS in the support set provided by each audio. The adaptive window length setting is shown in Table IV. After the introduction of adaptive window length, we increased the overall F-score by 1.4% and the F-score of HB by 18.6%. The experiment result is shown in Exp No.5 of Table III, which is the best result of our segment-level framework.

2) Frame-Level Embedding Learning: we adopt a 5s window to segment the data and make predictions according to the frame-level unit. In the training stage, We perform a 20-classification task using the Adam optimizer [32]. LR is 1e-4, and the StepLR is used with gamma 0.5 and step_size 10. CE loss is adopted, and the loss of overlap which is labeled 0 will not be calculated. The number of iterations is 100.

In the prediction stage, we define a 2-classification task. The 2-classifier is respectively initialized by calculating the 0-1 ratio of support set. Only the last two layers of the encoder and the 2-classifier are trained. Among them, the LR are 1e-4 and 1e-3, respectively. The number of iterations is 100. The result is shown in Exp No.6 of Table III. Compared with the best result of the segment-level framework which is shown in Exp No.5, the frame-level framework achieves an increase of 10.4% and 4.2% on ME and PB respectively, and an increase of 0.4% on the total F-score.

3) Semi-Supervised Learning: We set the threshold of pseudo-label 1 by counting the number of POS frames predicted correctly in the support set, and frames in the query set greater than and less than a threshold are then marked as 1 and 0, respectively. However, through the experiments, we find that we can get a better result by labeling 0 for those smaller than the threshold minus 0.2. In addition, we find that only fine-tuning the support set before the 86 iterations and adding semi-supervision after the last 14 iterations can achieve a better result. The experiment result is shown in Exp No.7 of Table III.

4) Weight Sharing Transfer Learning: As mentioned above, we mix the source domain and the target domain and define a 20-classification task. We make the 20-classifier and the 2-classifier trained together, meanwhile, and share the weight of class 1 of the 2-classifier and the weight of class 0 of the 20-classifier. Adam optimizer is used in the 20-classifier, and LR is set to 1e-3. We fine-tune the 20-classifier 100 iterations. Finally, we get our best performing system, as shown in Exp No.8 of Table III.

Observing that our frame-level system does not obtain the best result on HB, we analyse that since HB is dominated by mosquito sounds, and the frequency of mosquito sound is low, which leads to many short mute segments in a long-term POS segment. In response to this problem, the adaptive window length can judge the class of the entire window according to some effective segments in the window, while the frame-level system is prone to event truncation problems, resulting in a decrease in HB detection effect. We have tried the optimization

TABLE III

PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THE DCASE 2022 TASK5 VALIDATION SET, INCLUDING TRANSDUCTIVE INFERENCE-MUTUAL LEARNING/TRANSDUCTIVE INFERENCE (TI-ML/TI), DATA AUGMENTATION (DATA AUG), NEGATIVE SAMPLE SELECTION OPTIMIZATION (NEG), ADAPTIVE WINDOW LENGTH AND FIXED WINDOW SHIFT SCHEME (ALFS), SEGMENT-LEVEL/FRAME-LEVEL MODEL (SEGMENT/FRAME), SEMI-SUPERVISED LEARNING (SSL) AND TRANSFER LEARNING (TL).

Exp No.	Methods							Validation Set F-score			
	TI-ML/ML	Data Aug	Neg	ALFS	Segment/Frame	SSL	TL	HB	ME	PB	Total
1(Baseline)	TI-ML	✗	✗	✗	Segment	✗	✗	46.0	51.3	33.9	42.4
2	TI-ML	✓	✗	✗	Segment	✗	✗	70.3	55.0	37.3	50.7
3	TI	✓	✗	✗	Segment	✗	✗	66.6	78.4	52.3	63.9
4	TI	✓	✓	✗	Segment	✗	✗	67.3	81.9	52.7	65.2
5	TI	✓	✓	✓	Segment	✗	✗	**85.9**	79.3	48.1	66.6
6	TI	✓	✓	✗	Frame	✗	✗	69.0	89.7	52.3	67.0
7	TI	✓	✓	✗	Frame	✓	✗	80.7	87.6	51.4	69.3
8	TI	✓	✓	✗	Frame	✓	✓	77.0	**90.0**	**53.7**	**70.2**

TABLE IV

ADAPTIVE WINDOW LENGTH AND FIXED WINDOW SHIFT SCHEME. THE "//" DENOTES ALIQUOTING.

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

method of median filtering, and although it has achieved a better result, it can not completely solve this problem.

C. Results Analysis

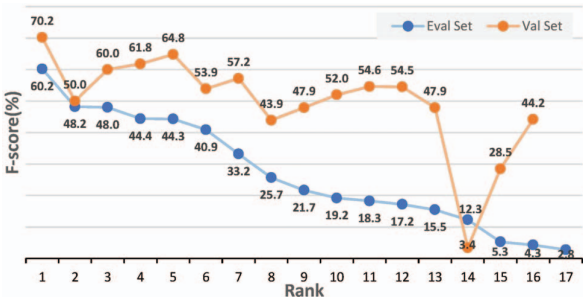
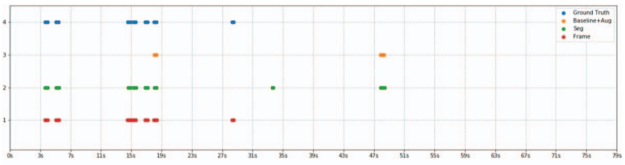


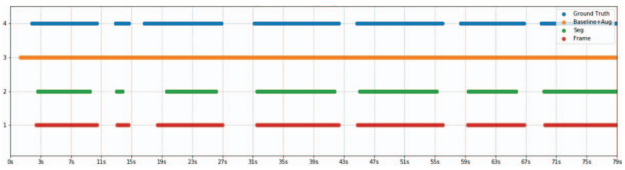
Fig. 2. DCASE 2022 Task5 F-score results for each team (best scoring system) on the evaluation and validation sets. Systems are ranked according to the F-score on the evaluation set. Ranked 1st is our best frame-level system. Ranked 14th and 15th are the official baseline systems based on template matching and prototypical network, respectively [2].These results and technical reports for the submitted systems can be found on the Task 5 results page 1.

1) Performance Comparison: There are 15 teams participating in DCASE 2022 Task5. Fig. 2 shows the results of all teams and two baseline systems on the validation and evaluation sets. It can be seen that our frame-level system ranks 1st in both validation and evaluation sets. The 3rd place team of used an event-length adapted ensemble of prototypical networks [18]. The 2nd place team adopted segment-level modeling and used AudioSet as additional data to expand the training data [19]. Our frame-level system achieves 10.2% and 12.0% higher F-scores on the validation set and evaluation set than the best results of the 2nd and 3rd teams, respectively. It is obvious that our method has competitive performance on the FSBED task.

2) Visualization and Analysis: To better illustrate the impact of different approaches, we selected two audio recordings and presented the corresponding FSBED results in Fig. 3. As can be seen, compared with Ground Truth (blue line), although the system using data augmentation can detect some sound events, there are obviously missed detections (such as ME1.wav) and false detections (such as R4_cleaned recording_TEL_20-10-17.wav). We obtained our best segment-level system (i.e., Exp No.5) after removing the ML framework, changing negative sample selection method and sliding window strategy. Compared with the Baseline+Aug system, it can reduce the occurrence of missed detection and false detection. As we improve the modeling accuracy, our final frame-level system (i.e., Exp No.8) can further improve the detection accuracy and get closer to Ground Truth.



(a) ME1.wav



(b) R4_cleaned recording_TEL_20-10-17.wav

Fig. 3. The visualization and comparison of FSBED results of different methods, including data augmentation used on baseline system (denoted as Baseline+Aug), the best segment-level system (denoted as Seg), and the best frame-level system (denoted as Frame).

IV. Conclusions

In this paper, we propose a frame-level embedding learning framework for FSBED, which improves the accuracy of detecting different bioacoustic events. Our method won the 1st place in DCASE 2022 task 5. In the future, considering that different types of bioacoustic events are quite different, more effective fine-tuning strategies can be explored to make the model more adaptable to detect new bioacoustic events. In addition, pre-training and more data augmentation methods can be tried to improve the generalization of the model. As the performance of the system is further improved, the FSBED system can better help biologists annotate more animal recordings at the lowest possible time cost to monitor biodiversity and animal behavior.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 62171427.

References

[1] Veronica Morfi, Inês Nolasco, Vincent Lostanlen, Shubhr Singh, Ariana Strandburg-Peshkin, Lisa F Gill, Hanna Pamuła, David Benvent, and Dan Stowell, "Few-shot bioacoustic event detection: A new task at the dcase 2021 challenge," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, 2021.

[2] I Nolasco, S Singh, E Vidana-Villa, E Grout, J Morford, M Emmerson, F Jensens, H Whitehead, I Kiskin, A Strandburg-Peshkin, et al., "Few-shot bioacoustic event detection at the dcase 2022 challenge," *arXiv preprint arXiv:2207.07911*, 2022.

[3] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[4] Çağdaş Bilen, Giacomo Ferroni, Francesco Tuveri, Juan Azcarreta, and Sacha Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 61–65.

[5] Annamaria Mesaros, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen, "Sound event detection in the dcase 2017 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, 2019.

[6] Annamaria Mesaros, Toni Heittola, and Anssi Klapuri, "Latent semantic analysis in sound event detection," in *2011 19th European Signal Processing Conference*. IEEE, 2011, pp. 1307–1311.

[7] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Tut database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[8] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon, "Sound event detection in synthetic domestic environments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 86–90.

[9] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 771–775.

[10] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[11] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[12] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu, "Instance credibility inference for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12836–12845.

[13] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua, "Interventional few-shot learning," *Advances in neural information processing systems*, vol. 33, pp. 2734–2746, 2020.

[14] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[15] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song, "Metagan: An adversarial approach to few-shot learning," *Advances in neural information processing systems*, vol. 31, 2018.

[16] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," *Advances in neural information processing systems*, vol. 31, 2018.

[17] Jack LeBien, Ming Zhong, Marconi Campos-Cerqueira, Julian P Velev, Rahul Dodhia, Juan Lavista Ferres, and T Mitchell Aide, "A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network," *Ecological Informatics*, vol. 59, pp. 101113, 2020.

[18] John Martinsson, Martin Willbo, Aleksis Pirinen, Olof Mogren, and Maria Sandsten, "Few-shot bioacoustic event detection using an event-length adapted ensemble of prototypical networks," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, 2022.

[19] Haohe Liu, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley, "Segment-level metric learning for few-shot bioacoustic event detection," *arXiv preprint arXiv:2207.07773*, 2022.

[20] Miao Liu, Jianqian Zhang, Lizhong Wang, Jiawei Peng, Chenguang Hu, Kaige Li, Jing Wang, and Qiuyue Ma, "Bit srcb team's submission for dcase2022 task5 - few-shot bioacoustic event detection technical report," Tech. Rep., DCASE2022 Challenge, 2022.

[21] Radoslaw Bielecki, "Few-shot bioacoustic event detection with prototypical networks, knowledge distillation and attention transfer loss," Tech. Rep., DCASE2021 Challenge, 2021.

[22] Hao Cheng, Chenguang Hu, and Miao Liu, "Prototypical network for bioacoustic event detection via i-vectors," Tech. Rep., DCASE2021 Challenge, 2021.

[23] Jens Johannsmeier and Sebastian Stober, "Few-shot bioacoustic event detection via segmentation using prototypical networks," Tech. Rep., DCASE2021 Challenge, 2021.

[24] Yue Zhang, Jun Wang, Dawei Zhang, and Feng Deng, "Few-shot bioacoustic event detection using prototypical network with background classs," Tech. Rep., DCASE2021 Challenge, 2021.

[25] Dongchao Yang, Helin Wang, Yuexian Zou, Zhongjie Ye, and Wenwu Wang, "A mutual learning framework for few-shot sound event detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 811–815.

[26] Mark Anderson and Naomi Harte, "Bioacoustic event detection with prototypical networks and data augmentation," *arXiv preprint arXiv:2112.09006*, 2021.

[27] Ren Li, Jinhua Liang, and Huy Phan, "Few-shot bioacoustic event detection: Enhanced classifiers for prototypical networks," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, 2022.

[28] Vincent Lostanlen, Justin Salamon, Andrew Farnsworth, Steve Kelling, and Juan Pablo Bello, "Robust sound event detection in bioacoustic sensor networks," *PloS one*, vol. 14, no. 10, pp. e0214168, 2019.

[29] Sinnojialin Pan, James Tin Yau Kwok, and Qiang Yang, "Transfer learning via dimensionality reduction," in *Proceedings of the National Conference on Artificial Intelligence*, 2008, vol. 2, p. 677.

[30] Dong-Hyun Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, p. 896.

[31] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[32] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.