FEW SHOT BIOACOUSTIC DETECTION BOOSTING WITH FINE TUNING STRATEGY USING NEGATIVE-BASED PROTOTYPICAL LEARNING

Technical Report

Yuna Lee, HaeChun Chung, JaeHoon Jung

KT Corporation, Republic of Korea

ABSTRACT

Few-shot sound event detection has always faced the challenge of detecting bioacousic sound events with only a few labelled instances of the class of interest. In this technical report, We describe our submission system for DCASE2023 Task5: few-shot bioacoustic event detection. We propose a novel framework of training audio segments via contrastive learning and prototypical learning, building the network more robust to the variety of acoustic environments, even in unseen domains. In addition, a finetuning strategy based on the novel loss functions is introduced. Our final systems achieves an f-measure of 83.08 on the DCASE task 5 validation set, outperforming the baseline performance and last year's first place by a large margin.

Index Terms— Few-shot Learning, Contrastive Learning, finetuning, bioacoustic sound Event Detection

1. INTRODUCTION

Sound event detection is the task of recognizing the sound events and their respective temporal start and end times in a recording [1]. In the case of bioacoustic sound event detection, the task focuses on animal vocalizations, which demand time and resources to annotate each time stamp [2]. All of these tasks meet the problem of data scarcity and difficulty in creating a robust model that can show general performance in acoustic domain. Accordingly, methods based on few-shot learning have come into the limelight. Few-shot learning (FSL) is a supervised learning method that can achieve high performance on data from completely different domains even with a small amount of data. In the previous DCASE 2022 task 5 challenge, submitted systems achieved great performance by transductive inference method [3, 4, 5], improved prototypical learning [6], contrastive learning [7], and multi-class classification learning via splitting the audio segment into frame-level [8]. Nevertheless, proposed methods showed relatively low performance on the evaluation dataset compared to the performance obtained on the validation set.

The majority of existing methods adopted prototypical learning to identify positive class from negative classes. Although prototypical learning itself demonstrated high performance, there were two limitations on taking the performance to another level. Firstly, the capability of high-level feature learning was challenging since the model was training on classifying binary classes, which are positive class and negative class. There has been an attempt to overcome this limitation by including additional multi-classification task along with existing few shot learning [8]. Second, the loss function of current prototypical learning [9] focuses on pulling positive classes, which we refer as "positive-based prototypical loss function (PPL)".

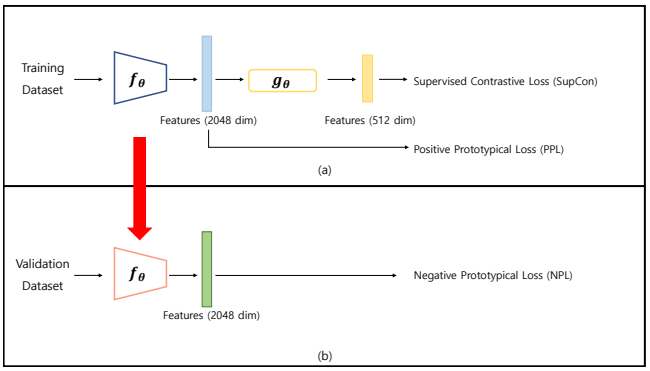

Figure 1: Our overall framework.

It may be promising on the training dataset which contains sufficient amount of positive class data, but it can lead to overfitting when the amount of negative class data is far much grater than that of positive class. If the model is trained on the standard prototypical learning manner, embedding features of negative classes are highly likely to be dispersed while that of positive classes are well-clustered in the embedding space. Since the class imbalance problem is pretty common in bioacoustic domain, we propose fine-tuning strategy with negative-based prototypical loss function (NPL) to ameliorate this issue. The proposed method suggests additional training on negative class data to enhance the ability to aggregate negative classes in the embedding space. By applying proposed strategy, the pretrained model can attain superior capability to discriminate positive classes and negative classes. Through this strategy, pretrained model can achieve higher F-measure on the validation dataset.

The rest of our paper is organized as follows. In the section 2, we describe our baseline framework and its novelties into specific details. In section 3, we outline the dataset and experimental setup for comparison not only with baseline methods, but also other variants of our own baseline framework. Experiment results are discussed in section 4, and we summarize our methods and argue future works in the section 5.

2. METHODS

Our 2-stage framework consists of pretraining stage and finetuning stage. Our overall framework can be shown in Fig. 1. Our framework achieves the best F-measure score 64.31% on the pretraining stage, and it can further be improved to F-measure score 83.08% after fine-tuning stage.

2.1. Outline

We utilize our system in N-way K-shot task. Prior to previous methods [3, 4, 5, 6, 7, 8], we denote positive segment as the target sound event and negative segment as the audio segments that do not contain the target sound event in each audio file. Given the fact that training dataset contains 45 classes and task 5 is regarded as 5-shot learning problem, we set $N = 45$ and $K = 5$. Following the rule that each audio file in the validation dataset should be considered independently, we define negative segments as negative classes. Instead of grouping negative segments into a single 'unknown' class, we define negative segments from a single audio file as solitary negative classes. In other words, each audio file contains a single positive class and single negative class. Alas, our system has 45 negative classes along with 45 positive classes. This enables encoder network $f_\theta(\cdot)$ to cluster positive segment more densely, maximizing the gap between positive segment and negative segments.

2.2. Pretraining Stage

In the pretraining stage, we train the encoder network $f_\theta(\cdot)$ by combining the advantages of prototypical learning and contrastive learning. We select each $2 \times K$ positive segments and negative segments from the dataset, and set K segments as support segments and the other as query segments. We denote the positive support set of class i as S_i^p and the query set as Q_i^p, and the negative support set and the query set of class i can be expressed as S_i^n, Q_i^n where $|S| = |Q| = K$. Since the prototype of each set is the mean embedding vectors, we can define the prototype of each set in class i as the equation below.

$$s_i^* = \frac{1}{|S_i^*|} \sum_{(x_i, y_i) \in S_i^*} f_\theta(x_i), q_i^* = \frac{1}{|Q_i^*|} \sum_{(x_i, y_i) \in Q_i^*} f_\theta(x_i) \quad (1)$$

where (x_i, y_i) are the segment and its label of the class i in each set. Equation 2 describes PP_j^i, which defines the euclidean distance between positive embedding vectors of Q_i^p and positive support prototype of class j, s_j^p.

$$PP_j^i = \left(\sqrt{\sum_{x \in Q_i^p} (f_\theta(x) - s_j^p)^2} \right) \quad (2)$$

In the same way, we can denote PN_j^i, which is the euclidean distance between embedding vectors of Q_i^p and negative support prototype s_j^n.

$$PN_j^i = \left(\sqrt{\sum_{x \in Q_i^p} (f_\theta(x) - s_j^n)^2} \right) \quad (3)$$

Then we can formulate positive-based loss for class i as the equation below.

$$ppl_i = -log \left(\frac{exp\left(-PP_i^i\right)}{\sum_{j=1}^{N} \left(exp\left(-PP_j^i\right) + exp\left(-PN_j^i\right)\right)} \right) \quad (4)$$

Using equation 4, PPL can be formulated as the equation 5.

$$PPL = \frac{1}{N} \sum_{i=1}^{N} ppl_i \quad (5)$$

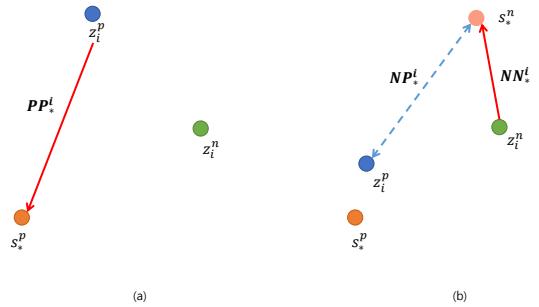

Figure 2: Let $z_i^p = f_\theta(x_i^p)$ be the positive embedding vector in the query set of class i, and $z_i^n = f_\theta(x_i^n)$ as the negative embedding vector in the following set. (a) depicts PPL function, which seeks to minimize PP_*^i. (b) describes the NPL function, minimize NN_* while maximize NP_*^i. Given that the encoder network already possesses the capability to cluster positive classes, we utilize NPL during the fine-tuning stage to increase the distance between s_*^n and s_*^p. The red line infers pull force, and the blue dotted line refers to push force.

To enlarge the feature representation learning We pretrain $f_\theta(\cdot)$ with PPL function and supervised contrastive loss function To enhance the feature representation cpapcity of $f_\theta(\cdot)$, we add supervised contrastive (SupCon) loss function [10]. We build 2-layer projection layer $g_\theta(\cdot)$ for creating embedding vectors for each audio segments in the following step. Thus, our total loss function for pretraining step can be formulated as $\mathcal{L}_{train} = \mathcal{L}_{PPL} + \mathcal{L}_{SupCon}$. We adopt convolutional neural network (CNN) from previous years' method [3] as our encoder network $f_\theta(\cdot)$. We set output embedding dimension to 2048 for $\mathcal{L}_{PPL}$, and downsize the dimension to 512 for $\mathcal{L}_{SupCon}$. Through the pretraining stage, the encoder network $f_\theta(\cdot)$ can attain the ability to embed positive classes well in the embedding space. In other words, the encoder network are taught in a way to focus on postiive-based feature learning during the pretraining stage.

2.3. Finetuning Stage

After first stage, $f_\theta(\cdot)$ is capable of detecting positive segment from negative segment. However, the dataset is comprised of a large number of negative segments and a very small amount of positive segments in the boiacoustic domain. This fact may not guarantee the sufficient performance of $f_\theta(\cdot)$ on the general bioacoustic domain. In order to resolve data scarcity and performance maintenance issues, we figured that a sole training stage was not enough. Based on the unique characteristic of bioacoustic dataset, we finetune $f_\theta(\cdot)$ to aim on negative-based feature learning, which is opposite of the aforementioned stage. We display comparison of PPL and NPL in Fig. 2. Furthermore, We also propose a further developed Distance-based NPL fucntion by incorporating the idea of Farthest Point Sampling (FPS) algorithm into the NPL function proposed in this technical report.

Negative-based Prototypical Loss In this stage, we add additional definition of distances between embedding vectors of Q_*^n and support prototypes. Following the equations 2 and 3, we can define NP and NN as euclidean distance of negative query embedding vectors

between positive support prototype and negative support prototype. Equation 6 and 7 describes NP and NN in more specific manner. we can formulate NPL with equations 8 and 9.

$$NP_j^i = \left(\sqrt{\sum_{x \in Q_i^n} (f_\theta(x) - s_j^p)^2} \right) \tag{6}$$

$$NN_j^i = \left(\sqrt{\sum_{x \in Q_i^n} (f_\theta(x) - s_j^p)^2} \right) \tag{7}$$

Unlike the PPL, NPL minimize the distance between negative embedding vectors and s^n while maximizing the distance between the positive embedding vectors. Therefore, we redesign the positive-based loss ppl_i as the equation 8.

$$pnl_i = -log \left(\frac{exp\left(PN_i^i\right)}{\sum_{j=1}^{N} \left(exp\left(PP_j^i\right) + exp\left(PN_j^i\right)\right)} \right) \tag{8}$$

And we add new negative-based loss npl_i to minimize the gap between negative embedding vectors and s^n. The following distance function are described as below.

$$nnl_i = -log \left(\frac{exp\left(-NN_i^i\right)}{\sum_{j=1}^{N} \left(exp\left(-NP_j^i\right) + exp\left(-NN_j^i\right)\right)} \right) \tag{9}$$

To sum up, NPL function can be summarized as equation 10.

$$NPL = \frac{1}{N} \sum_{i=1}^{N} (pnl_i + nnl_i) \tag{10}$$

By finetuning $f_\theta(\cdot)$ with $\mathcal{L}_{NPL}$, $f_\theta(\cdot)$ learns the ability to cluster negative embedding vectors and negative prototype more densely and gives the effect of separating positive segments in result.

Distance-based Negative-based Prototypical Loss While NPL loss function randomly pick K support features and K query features from $2 \times K$ arbitrarily chosen features, we enlarge NPL loss function by adopting the idea of Farthest Point Sampling (FPS) algorithm. FPS algorithm is classic method used in 3D point clouds [11]. Since we aims to clump negative embedding vectors and negative prototype, we believe distance-based selection of query and support features can maximize the efficacy of NPL loss function. All distances between $2 \times K$ randomly extracted positive features and $2 \times K$ negative features are calculated. Then, the positive and negative features with the shortest distance are selected as a pair of reference features. Nearest sampling is attempted based on the selected positive reference feature and negative reference feature. Thus, we set negative features placed close to the positive features as a negative support set, and positive features closely located to the negative features as a positive query set. Then, we optimize the loss function to maximize the distance between negative prototype and positive query set so that we can ultimately maximize PN. In opposite manner, we conduct furthest sampling based on the prior negative reference feature in negative features. By this process, negative features located on the outskirts will be selected from among negative features, and non-selected features will be located on the inner side among negative features. We set the selected features to a negative query set and the unselected features to a negative support set. The negative prototype created from negative support set are used to minimize the distance between negative query set, eventually minimizing NN. In this way, we can boost the initial goal of NPL by optimizing the maximization of positive-negative distance and minimization of negative-to-negative distance at the same time. The following procedures are specifically illustrated in the Figure 3.

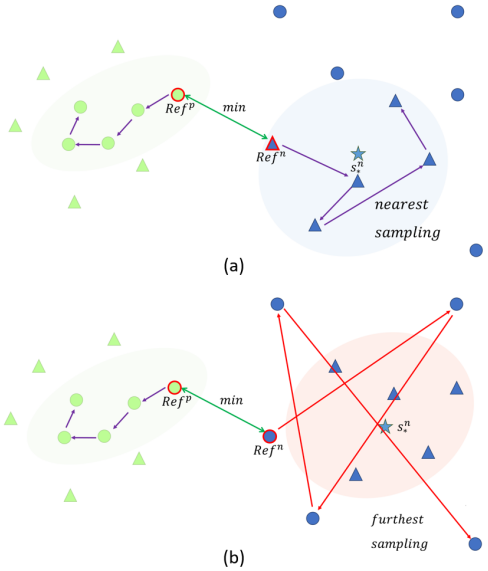

Figure 3: We denote each positive reference feature negative reference feature as Ref^p and Ref^n. The triangle, circle, and star-shaped figure each represent the feature vectors of support set, the query set, and the prototype respectively. (a) shows the process of maximizing PN by obtaining a positive query set and a negative support set close to each Ref^* through nearest sampling. (b) is the process of minimizing the NN between the negative support set and the query set through furthest sampling.

2.4. Post-processing & Inference

For post-processing and inference, We applied methods proposed on previous year's challenge [3].

3. EXPERIMENT

3.1. Experimental setup

In DCASE 2022 challenge, most of the methods were developed based on the transductive inference (TI) method [12, 13, 14], which played a crucial role on winning the DCASE 2021 challenge [15]. Based on this experience, we applied part of the TI method as a variant to our system. We conducted the experiments for two reasons. First, we want to prove that our novel framework is more applicable in the few shot learning domain than baseline methods. Further, we compare variants with our baseline as ablation study to analyze the impact of TI methods and our novel finetuning strategies. Second, We intend to prove the efficacy of our proposed method by comparing the results of grafting the finetuning strategy onto the existing baseline method. In all experiments, the learning rate was set to 0.001 and the input length was fixed in 0.2 seconds.

		PB			ME			HB			Overall		
Stage	System	Pre (%)	Rec (%)	F-measure (%)	Pre (%)	Rec (%)	F-measure (%)	Pre (%)	Rec (%)	F-measure (%)	Pre (%)	Rec (%)	F-measure (%)
Pretrain	Jung_S0	52.91	39.57	45.27	66.67	84.62	74.58	88.61	74.02	80.66	66.39	59.28	62.64
	Jung_S1	61.54	38.26	47.18	84.91	86.54	85.71	80.86	65.71	72.50	74.27	56.70	64.31
	Jung_S2	52.91	39.57	45.27	66.67	84.62	74.58	88.61	74.02	80.66	66.39	59.28	62.64
	Jung_S3	61.54	38.26	47.18	84.91	86.54	85.71	80.86	65.71	72.50	74.27	56.70	64.31
Finetune	Jung_S0	77.30	47.39	58.76	96.30	100.00	98.11	96.79	95.77	96.28	89.15	72.22	79.79
	Jung_S1	79.47	52.17	62.99	91.23	100.00	95.41	96.80	95.92	96.36	88.56	75.77	81.67
	Jung_S2	84.06	50.43	63.04	91.22	100.00	95.41	96.04	95.17	95.60	90.17	74.38	81.52
	Jung_S3	76.22	54.35	63.45	98.11	100.00	99.05	99.53	95.62	97.53	89.93	77.20	83.08

Table 1: The precision, recall, and f-measure of each subset in the validation set. S0, S1,S2,S3 are four systems we submitted to the challenge.

To prevent overfitting on any dataset, we implemented early stopping. We did not use any augmentation or additional acoustic features. We adopted the official evaluation metric[1] as our evaluation metric.

3.2. Dataset

The DCASE 2023 task 5 dataset contains a training set, a validation set, and an official evaluation set. Since the full annotation of evaluation set was not released in public, we considered the validation set of the DCASE 2023 task 5 dataset as the evaluation set.

4. RESULTS

4.1. Performance Comparison

In Table 1, we compare our submitted systems. We submit 4 systems in the challenge. We select systems with different conditions as mentioned in Section 3.1 to avoid cherry-picking models that might overfit on the validation set. All four systems use the same configurations shown in Table 2, and the performance is specified in Table 1. When it comes to specifically comparing the performance between sub-folders, our system showed relatively low performance on PB dataset relative to other dataset in sub-folders. We assume this phenomenon is due to the drastic ratio between positive segment and negative segment. Unlike other datasets, PB dataset contains relatively short duration of positive segment. This fact assures PB dataset is comprised of highly imbalanced ratio of positive segments to negative segments. Since the features extracted from positive segments are limited, the encoder network $f_\theta(\cdot)$ finds it more difficult to detect positive semgents.

		Precision (%)	Recall (%)	F-measure (%)
Template Matching		2.42	18.32	4.28
Prototypical Network		36.34	24.96	29.59
[8]		77.50	71.50	74.40
Ours	Pretraining	74.27	56.70	64.31
	Finetuning	89.93	77.20	83.08

Table 2: The precision, recall, and measure of validation set.

In the Table 2, we compare our methods with baseline methods and the winning team of DCASE 2022 [8]. The baseline methods are template matching and prototypical network [16]. Pretraining denotes the performance of the encoder $f_\theta(\cdot)$ after the pretraining stage, and Finetuning denotes the performance after the finetuning stage. As can be seen in Table 2, our proposed method outnumber both baseline methods and 2022 challenge winning team by large margin. We also evaluated our encoder network $f_\theta(\cdot)$ after each stage to confirm the impact of NPL function. The disparity between the performance of two stages clearly verify NPL function actually have a meaningful impact on developing the capacity to detect positive sound event even in the highly imbalanced dataset.

4.2. Ablation Study

In the ablation study, we compare our baseline framework and the combination of baseline and novel finetuning strategy. We compared the case where only the basic training stage was performed for each baseline and the case where two different finetuning strategies were applied.

	Train set	**w.** validation set
Pretraining stage	62.64	64.31
w. NPL finetuning	79.79	81.52
w. Distance-based NPL finetuning	81.67	83.08

Table 3: Ablation study of the proposed method.

In the case of the dataset used for training, it was divided into a case where only the training set was used and a case where the validation set was used audio file-wise based on the TI method. As the Table 3 shows, It is clear that our method excels if finetuning strategy is applied. The fact that finetuning strategy with NPL function and Distance-based NPL function show noticeable numerical difference is also noteworthy. As we do not have full annotation of evaluation dataset, we could not compare the final F-measure score of each system. Alternatively, we extracted t-SNE [17] from evaluation set. When we associate the t-SNE from each system, we were able to confirm that the Distance-based NPL function works effectively in embedding positive segments and negative segments in different space.

5. DISCUSSION

In this technical report, we present novel framework for few-shot bioacoustic event detection. Our method combine contrastive learning method and prototypical learning, and use novel finetuning strategy of using modified prototypical loss function. While pretraining process enable embedding positive class data on the embedding space, NPL finetuning strategy enable pretrained network to detect sound events in the environment where positive sound events were unseen in the training stage or fine-tuning stage. Thus, We claim that our finetuning strategy can robustly separate positive and negative segments even in highly imbalanced datasets.

[1] https://github.com/c4dm/dcase-few-shot-bioacoustic

6. 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] http://dcase.community/challenge2023/.

[3] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey system for dcase 2022 task 5 : Few-shot bioacoustic event detection with segment-level metric learning technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[4] Y. Tan, L. Xu, C. Zhu, S. Li, H. Ai, and X. Shao, "A new transductive framework for few-shot bioacoustic event detection task," June 2022.

[5] Q. Huang, Y. Li, W. Cao, and H. Chen, "Few-shot bio-acoustic event detection based on transductive learning and adapted central difference convolution," June 2022.

[6] D. Yang, Y. Zou, F. Cui, and Y. Wang, "Improved prototypical network with data augmentation," June 2022.

[7] B. Zgorzynski and M. Matuszewski, "Siamese network for few-shot bioacoustic event detection," June 2022.

[8] J. Tang, X. Zhang, T. Gao, D. Liu, J. P. Xin Fang and, Q. Wang, J. Du, K. Xu, and Q. Pan, "Few-shot embedding learning and event filtering for bioacoustic event detection," June 2022.

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

[10] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[12] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Piantanida, and I. Ben Ayed, "Information maximization for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2445–2457, 2020.

[13] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 979–13 988.

[14] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.

[15] D. Yang, H. Wang, Z. Ye, and Y. Zou, "Few-shot bioacoustic event detection= a good transductive inference is all you need," DCASE2021 Challenge, Tech. Rep, Tech. Rep., 2021.

[16] V. Morfi, I. Nolasco, V. Lostanlen, S. Singh, A. Strandburg-Peshkin, L. F. Gill, H. Pamula, D. Benvent, and D. Stowell, "Few-shot bioacoustic event detection: A new task at the dcase 2021 challenge." in *DCASE*, 2021, pp. 145–149.

[17] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.