SURREY SYSTEM FOR DCASE 2022 TASK 5: FEW-SHOT BIOACOUSTIC EVENT DETECTION WITH SEGMENT-LEVEL METRIC LEARNING

Technical Report

Haohe Liu[1]*, Xubo Liu*[1]*, Xinhao Mei*[1]*, Qiuqiang Kong*[2]*, Wenwu Wang*[1]*, Mark D. Plumbley*[1]

[1] Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, UK
[2] Speech, Audio, and Music Intelligence (SAMI) Group, ByteDance, China

ABSTRACT

Few-shot audio event detection is a task that detects the occurrence time of a novel sound class given a few examples. In this work, we propose a system based on segment-level metric learning for DCASE 2022 challenge few-shot bioacoustic event detection (task 5). We make better utilization of the negative data within each sound class to build the loss function, and use transductive inference to gain better adaptation on the evaluation set. For the input feature, we find the per-channel energy normalization concatenated with delta mel-frequency cepstral coefficients to be the most effective combination. We also introduce new data augmentation and post-processing procedures for this task. Our final system achieves an f-measure of 68.74 on the DCASE task 5 validation set, outperforming the baseline performance of 29.5 by a large margin. Our system is fully open-sourced[1].

Index Terms— few-shot learning, transductive inference, metric learning, audio event detection

1. INTRODUCTION

Few-shot learning (FSL) [1] is a machine learning problem that aims to generalize the system to novel classes after supervised learning with the data from known classes. Sound event detection [2] is a task that locates the onset and offset of certain sound classes. By combining the idea of FSL with sound event detection [3], the system can detect the new sound class with only a few labeled samples. This technique is useful for audio data labeling, especially when the user needs to detect a new type of sound.

In the previous DCASE 2021 task 5 challenge, the submitted systems [4, 5, 6, 7] have achieved promising results on the evaluation set. Most of the systems use a prototypical network [1] as the main architecture. Yang et al. [4] propose a mutual learning framework to improve the feature extractor. Tang et al. [5] propose to use embedding propagation to learn a smoother manifold in the latent space. Data augmentations such as spec-augment and mixup are used in the system described in [6, 7].

In previous works, the negative segment in each audio file is not fully utilized. Here negative segment means the audio segments that do not contain the target sound event. Most metric-learning-based studies [4, 5] optimize the model only using the labeled positive data, by grouping and separating the latent prototypes of the same and different positive classes, respectively. However, the positive segments also need to be distinguishable from the negative segments within the same audio file. We propose to use negative segments to enhance the metric learning to address this problem.

This report describes the prototypical-network based system we submitted to the DCASE challenge 2022 task 5. Our system builds upon a prototypical network and proposes new methods for the input features, metric learning, evaluation data preprocessing, and post-processing.

To increase the generalization ability of our system, we merge the training set with the animal sound in AudioSet [8]. We conduct studies on a different combinations of audio features to choose the best feature for this task. Features includes log-mel spectrogram (MEL), per-channel energy normalization (PCEN), mel-frequency cepstral coefficients (MFCC), and delta-MFCC (ΔMFCC). We found using PCEN and delta-MFCC together yields the best average f-measure score on the validation set. To make full use of the labeled few-shot data, we integrate transductive inference into our metric learning framework. We treat each evaluation file as a sound class and jointly optimize it with the training dataset. Our proposed system achieves a f-measure of 68.74 on the DCASE task 5 validation set.

This technical report will be organized as follows. Section 2 provides an overview of our system. Section 3 introduces the methods we proposed to improve our system. Section 4 discussed the experimental setup and the four systems we submitted to the challenge. Section 5 reports the result of the validation set and the ablation studies. Section 6 summarize this work and provide a conclusion.

2. SYSTEM OVERVIEW

We build our system with a prototypical network, which is widely used for metric-based few-shot learning. The objective of our system is properly mapping different sounds into a latent embedding within a high-dimensional space, where similar sounds have a closer distance.

We use episodic training [9] to optimize our system in an N-way-K-shot way, in which we set $N = 10$ and $K = 5$. This means each training batch will select data from N classes. And for each class, the system will select $2 \times K$ segments from the dataset, and treat K segments as support segments and other K segments as query segments. Then both segments are processed by a feature extraction network, which outputs latent embeddings. By averaging the embedding of K support and K query segments for class i, we obtain a query prototype x_i^q and a support prototypes x_i^s. The system is optimized by minimizing the distance between query prototypes and support prototypes for each class. To build a robust latent space and to generalize better to the new class, we introduce the metric learn-

[1] https://github.com/haoheliu/DCASE_2022_Task_5

ing with negative samples and the transductive inference approach in Section 3.2 and 3.3.

During evaluations, only fivethe system uses K labeled positive events to build positive and negative prototypes, which are the latent representation of positive and negative segments in one audio file. Here a positive sound event can contain multiple positive segments. If the negative segment in the labeled part is too short, we will use a spectrum-similarity-based algorithm (Section 3.6) to identify the possible negative segment in the remaining unlabelled audio. To classify the unlabelled part, the system will first segment it with an adaptive segment length and build a query set. Then the query set can be classified by calculating the distance with prototypes. We introduce the way we build the positive and negative prototypes in Section 3.6.

Post-processing is important to the performance of the system. We introduce a splitting-merging-filtering-based post-processing method designed to fully utilize the information provided in the first K labeled events, such as minimum negative segment length. We also use various data augmentation techniques and external training data from AudioSet during model training, which will be discussed in section 4.

3. METHODOLOGY

3.1. Feature extraction network

Our feature extraction network f_θ is a convolutional neural network (CNN) based architecture that maps the audio feature s into a latent embedding x:

$$f_\theta : s \rightarrow x \tag{1}$$

Similar to the architecture proposed by [10], the network f_θ consists of three convolutional blocks with hidden channels of 64, 128, and 64. Each convolutional block consists of three two-dimensional CNN layers with batch normalization and leaky rectified linear unit activations [11]. To enlarge the reception field, we apply 2×2 max-pooling after each block. The input and output of each convolutional block have a residual connection processed by a downsampling CNN layer. In order to maintain the same output dimension with different input lengths, we apply an adaptive average pooling at the end of the network. The final output feature map after adaptive pooling is a $C \times T \times F$ size block, which is also the final latent embedding.

3.2. Segment-level metric learning

In a similar way to [3], we divide the audio features into segments with equal lengths for metric learning. Each segment has a unique label indicating its sound class and whether it is a positive or negative event. As illustrated in the left part of Figure 1, previous methods typically optimize the model only using the labeled positive classes, by minimizing the distance between the query and support prototypes that have the same label. However, these systems only use the labeled positive classes, which are limited in duration. By comparison, negative samples usually contain a wider variety of sounds and are easier to collect. In addition, during inference, the system will need to discriminate between positive and negative samples. This objective is not directly optimized during training with only positive data.

So, to better discriminate between negative and positive segments within the audio, we propose to perform a segment-level metric learning approach with the negative segments. In on the probabilities of the positive classes, the query will also need to calculate the distance with the negative support embeddings. The comparison between positive and negative segments will provide the model with more contrastive information on building the latent space. Note that we do not optimize the probabilities of the negative query prototypes, because negative samples are not guaranteed to have the same sound for classification.

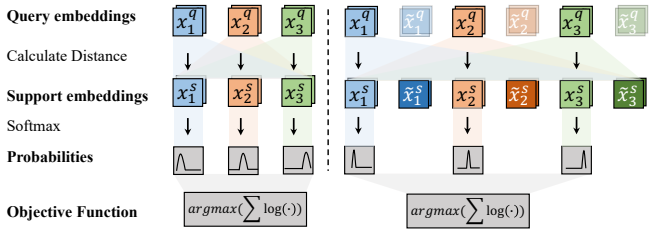

Figure 1: The left side is the baseline method that performs metric learning in a 3-way-2-shot way. The baseline method only uses positive classes. The right side illustrates the improved metric learning with the negative samples method we used in our system. x_i and $\widetilde{x}_i$ stand for the positive and negative samples of class i. Transparent color means the samples are not actually used.

3.3. Transductive inference

We adopt a transductive inference [12] approach during training. Specifically, our model will be optimized both on the development set and the partially labeled evaluation set. However, the labeled segment in the evaluation set is much shorter than in the training set. To balance between these two sets, we uniformly sample N sound class from both sets during training.

3.4. Loss function

We optimize f_θ by maximizing the probability that x_i^q belongs to class i. Specifically, we first calculate the probability distribution of x_i^q among N possible classes by Equation 2. Then we maximize the probability that x_i^q belongs to class i by summing and maximizing the diagonal elements of $\mathbf{D}$, given by

$$\mathbf{D}_{i,j} = \sqrt{\Sigma((x_i^q - x_j^s)^2)},\ \widetilde{\mathbf{D}}_{i,:} = \log(\text{Softmax}(\mathbf{D}_{i,:})), \tag{2}$$

$$l = \text{argmax}_\theta(\Sigma(\mathbb{I} \odot \widetilde{\mathbf{D}})), \tag{3}$$

where l is the loss function, $\mathbb{I}$ is the identity matrix, $\odot$ stands for element-wise multiplication, and Σ means summing all the elements in the matrix.

3.5. Adaptive segment length

During the inference process, the audio file will be divided into segments with adaptive segment lengths. The labeled positive segments will be used to build positive prototypes. The segments after the last labeled positive event, which are unlabeled segments, will be treated as the query set. Both the query set and labeled negative segments can be used to build the negative prototypes.

Sounds of different animal or bird species have drastically different lengths of vocalization, ranging from 30 milliseconds to 5 seconds. To have robust detection result, we set different window lengths and hop lengths for each audio file based on $t_{\max} =$

$t_{\max}$ (s)	[0,0.1]	(0.1,0.4]	(0.4,0.8]	(0.8,3.0]	(3.0,+inf)
Length	8	$t_{\max}$	$t_{\max}$ / 2	$t_{\max}$ / 4	$t_{\max}$ / 8

Table 1: The window length we use during segmenting the evaluation and validation audio file for different values of $t_{\max}$.

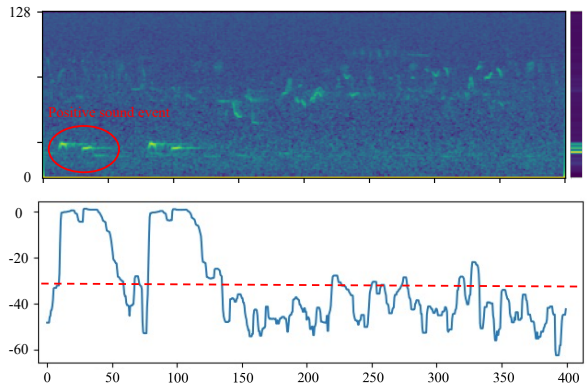

Figure 2: The visualization of the negative sample searching algorithm. In the first row, the left picture is the log-mel spectrogram of an audio clip, and the picture on the right side is the mel-frequency bins weight. The second row shows the frequency pattern matching score. The segment with a score under the red dashed line will be used as negative segment.

$\max(t_1, ..., t_K)$, as shown in Table 3.5, where $t_1, ..., t_K$ denotes the duration of the K labelled positive events. We set the hop length as one-third of the window length.

3.6. Positive and negative prototypes

We calculate the positive prototype by simply averaging the embeddings of the labeled positive segments because we assume positive events do not contain too much variety. By comparison, building negative prototypes is more tricky for the following challenges:

1. Negative segments can contain many different kinds of sounds. So simply averaging all the negative embeddings will result in a sub-optimal representation of negative prototypes.
2. The duration of labeled negative data is limited in some test files. In this case, some negative events appear later in the query set may not be included in the negative prototype.

Note that the baseline system uses the average embedding of the entire audio as the negative prototype, under the assumption that the positive event is sparse. However, this assumption is not true in most of the evaluation files. Building a negative prototype in this way can lead to a degraded result.

To address the first challenges, we choose to run our evaluation six times, each with different 30 randomly selected negative segment embeddings. And we mean the predicted probabilities across time as the final predictions. In this case, the negative prototype in each run can have a chance to represent different sound classes.

For the second challenge, we propose a negative sample searching algorithm (Figure 3.6) to find more negative segments to build robust negative prototypes. The algorithm includes a frequency bins weighting step and a frequency pattern matching step.

The frequency bins weighing operation aims to find the frequency band that is most likely to contain the target sound event so that we can find the negative event more accurately. This operation is based on two assumptions. First, we assume that bioacoustic sound events for a certain class (e.g., owl vocalization) do not have too many variations in frequency. Localizing the most important frequency band can therefore assist the model to ignore interference sounds in other frequency bands. Second, since onset and offset time mark the start and end of a sound event, we assume that the frequency bins that have more energy fluctuations before and after one positive event are the frequency bands that contain target sound events. We calculate frequency bin weight w by calculating the energy difference between the positive segments and the negative segments before and after these events, and then w is normalized between zero and one.

Then we perform frequency pattern matching to locate possible negative samples. We first multiply the linear mel-spectrogram with the frequency bin weights. Then we calculate the average of positive time steps as the frequency pattern template. Finally, we calculate the scale-invariance signal-to-noise ratio (SISNR) [13] between the template and each time step in the weighted mel-spectrogram. If the result of a time step is smaller than a threshold, we will treat it as a negative event. The threshold is calculated using the minimum SISNR value between the template and positive segments. Here we use the scale-invariance version of SNR to overcome the situation where the energy of a positive event changes.

3.7. Post processing

We determine the post processing strategy of a sound class based on maximum length of positive event $t_{\max}$ and minimum length of negative event $\widetilde{t}_{\min} = \min(\widetilde{t}_1, ..., \widetilde{t}_K)$, where $\widetilde{t}_1, ..., \widetilde{t}_K$ is the durations of the K labelled negative segments. We perform the following steps for post processing. Note that most parameters in the following steps are chosen by experience.

$\widetilde{t}_{\min}$ (s)	[0, 8]	(8, 100]	(100, +inf)
Length	8	$\widetilde{t}_{\min}/2$	100

Table 2: The window length we use during short negative segment detection for different values of $\widetilde{t}_{\min}$.

After detecting the onset and offset of short negative segments, we perform splitting on the positive segment. This operation is only performed when the length of the detected positive event is longer than $2t_{\max}$ and a negative segment is detected within the positive event.

We determine the threshold segment length of a sound class with the duration of the few labeled positive and negative segments $\mathbf{t}$ and $\mathbf{t}'$. During filtering, the minimal segment length is $0.4 * \bar{\mathbf{t}}$. The merging operation will be performed if two segments have a total duration less than $0.8 * \bar{\mathbf{t}}$ and the negative segment between them is shorter than the minimal duration in $\mathbf{t}'$. The system will consider to perform splitting operation if the segment length is longer than $2.0 * \bar{\mathbf{t}}$. And we design the negative segment mining approach to determine where to perform the splitting.

The essential reason why the model cannot locate the negative segment between two adjacent positive sound events is because of the length of the query window. There is a trade-off during inference. If the window length is not small enough, the query sample could contain positive and negative samples at the same time, and the negative part will not be effectively detected. However, if the

	HB			ME			PB			Overall		
System	Pre (%)	Rec (%)	F (%)	Pre (%)	Rec (%)	F (%)	Pre (%)	Rec (%)	F (%)	Pre (%)	Rec (%)	F (%)
Liu_S1	**98.87**	79.31	**88.01**	78.95	**86.54**	82.57	48.18	28.70	35.97	68.90	50.84	58.51
Liu_S2	87.03	**83.08**	85.01	65.00	50.00	56.52	36.13	30.00	32.78	54.99	45.89	50.03
Liu_S3	94.50	59.67	73.15	65.85	51.92	58.06	35.40	17.39	23.32	55.54	32.08	40.67
Liu_S4	97.95	79.46	87.74	86.27	84.62	85.44	57.52	27.66	37.36	**76.56**	49.54	60.16
Liu_S0	82.20	82.33	82.26	**95.56**	82.69	**88.66**	**59.88**	**42.17**	**49.49**	76.28	**62.56**	**68.74**

Table 3: The precision, recall, and f-measure of each subset in the validation set. S1,S2,S3,S4 are four systems we finally submitted to the challenge. We do not submit S0 on considering that the model might overfit to the validation dataset.

window length is too small, the model tends to have too many false positives that will affect the final score.

To address this problem, we propose negative segment mining operations, which means detecting the possible negative segment. Since the false positive is not important in this case, we can use a small query window length that is capable of detecting small intervals between two positive sound events. The detected negative intervals will be used for splitting long positive segments if the segment length is longer than $2.0 * \bar{\mathbf{t}}$.

4. EXPERIMENTS

4.1. Dataset

Challenge official dataset DCASE 2022 task 5 dataset contains a training set, a validation set, and an evaluation set. The training and validation set are both fully labeled. The evaluation set is provided only with the labels of the first five positive events.

AudioSet The training set of the DCASE 2022 task 5 challenge only contains 47 different sound classes. To enhance the latent space with a wider variety of sounds, we choose to build an extra training set using the strongly labeled part of the AudioSet dataset. To avoid the domain mismatch problem, we only utilize the sound labels that are related to animal vocalizations. We only use animal sounds that do not overlap with other non-animal sounds. We will not use the data if the total duration of non-overlapping parts is less than two seconds within the audio because we consider in this case the sound event is not enough to use.

4.2. Evaluation metric

We use the f-measure score, the official evaluation metric[2] provided by the organizers of DCASE task 5, as our main evaluation metric. We also report the accuracy and recall.

4.3. Experimental setup

Following the prior works [4], all the audio data are resampled into 22.5 kHz sampling rate. The input feature of our system is the stack of PCEN [14] and delta-MFCC [15] features. In the short-time Fourier transform, we set the window length as 1024 and hop size as 256. We set the mel dimension as 128. The input length of our model during training is 0.2 seconds, and the output embedding dimension is 2048. All the experiments use an initial learning rate of 0.001 with 0.65 exponential decay every 10 epochs. We will stop model training if the validation accuracy does not improve for 10 consecutive runs. And the model with the best validation accuracy is used for evaluation. For model evaluation, we consider the negative segments with a total duration of fewer than 2.0 seconds are not enough to build the negative prototype. And in this case, we will use the negative sample searching algorithm mentioned in Section 3.6.

Different from the baseline that preprocesses and segments the training data with a certain hop length, we implement a dynamic approach that generates training data on the fly. During training, we will randomly select the starting point of the training segment. In this case, the training data can be fully utilized.

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

5. RESULT

Method	Precision (%)	Recall (%)	F-measure (%)
Template Matching	2.42	18.32	4.28
Prototypical [16]	36.34	24.96	29.59
Proposed	76.28	62.56	68.74

Table 4: Comparison with baseline template matching and prototypical network methods

The performance of our system on the validation set is reported in Table 5. The F-measure score of the prototypical network baseline is 29.59. Our system outperforms the baseline by a large margin with an F-measure score of 68.74, using a threshold of 0.95. We submit four systems for the DCASE task 5 challenge. The performance of each system on each validation subset is reported in Table 3.7. We select systems with different performances to avoid cherry-picking models that only work well on the validation set. All four systems use the same configurations except the settings shown in Table 5. We tried different post-processing methods and training data in these four systems.

System	Pad	Split	Merge	AudioSet	Threshold	Ens
Liu_S1	✗	✓	✗	✗	0.5 - 0.995	✓
Liu_S2	✓	✗	✗	✓	0.6	✗
Liu_S3	✓	✓	✓	✗	0.95	✗
Liu_S4	✗	✗	✗	✗	0.6	✗

Table 5: The setting of each submitted system. Ens stand for performing ensamble on different thresholds (0.5,0.7,0.9,0.99,0.995).

6. CONCLUSIONS

This technical report describes the system we submitted to the DCASE 2022 challenge task 5. Our system use segment-level metric learning with negative segments, negative segment searching, and post-processing. The experimental result indicates our system can improve the baseline prototypical network by a large margin.

7. ACKNOWLEDGMENT

This research was partly supported by a PhD scholarship from the Centre for Vision, Speech and Signal Processing (CVSSP), Faculty of Engineering and Physical Science (FEPS), University of Surrey and BBC Research and Development, and a Research Scholarship from the China Scholarship Council (CSC) No. 202006470010. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

8. REFERENCES

[1] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] 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.

[3] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 81–85.

[4] 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., 2021.

[5] T. Tang, Y. Liang, and Y. Long, "Two improved architectures based on prototype network for few-shot bioacoustic event detection," Tech. Rep., 2021.

[6] Y. Zhang, J. Wang, D. Zhang, and F. Deng, "Few-shot bioacoustic event detection using prototypical network with background classs," DCASE2021 Challenge, Tech. Rep., 2021.

[7] M. Anderson and N. Harte, "Bioacoustic event detection with prototypical networks and data augmentation," *arXiv:2112.09006*, 2021.

[8] 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)*, 2017, pp. 776–780.

[9] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1446–1455.

[10] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[11] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv:1505.00853*, 2015.

[12] 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.

[13] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[14] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5670–5674.

[15] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for mfcc feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*, 2010, pp. 946–953.

[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.