# TWO IMPROVED ARCHITECTURES BASED ON PROTOTYPE NETWORK FOR FEW-SHOT BIOACOUSTIC EVENT DETECTION

Technical Report

*Tiantian Tang, Yunhao Liang, Yanhua Long*

Shanghai Normal University, Shanghai, China
1000479042@smail.shnu.edu.cn, winnerahao@163.com, yanhua@shnu.edu.cn

## ABSTRACT

In this technical report, we describe our submission system for DCASE2021 Task5: few-shot bioacoustic event detection. Few improvements are investigated to better the baseline of deep learning prototypical network. Including the $N$-way 5-shot classification prototypical network training strategy, data augmentation techniques, the proposed embedding propagation and attention similarity approaches. On the official validation set, we demonstrate that the proposed method achieves the overall F-measure score of $54.7\%$ on the validation set.

***Index Terms—*** Few shot learning, sound event detection, embedding propagation, attention similarity

## 1. INTRODUCTION

This report presents the technical details of our SHNU submission system for DCASE2021 Challenge Task5: few-shot bioacoustic event detection [1]. The challenge of this task is to find reliable algorithms that are capable of dealing with data sparsity, class imbalance and noisy/busy environments. Contrary to standard supervised learning paradigm, few-shot learning describes tasks in which an algorithm must make predictions given only a few instances of each class.

In this challenge, the few-shot task is required to run as a 5-shot task. Only five annotated calls from each recording in the evaluation set are provided to the participants. Each recording of the evaluation set have a single class of interest which the participants will then need to detect through the recording. Each recording can have multiple types of calls or species present in it, as well as background noise, however only the label of interest needs to be detected.

We submit four systems to the challenge, they are, 1) SHNU1: 3-way 5-shot prototypical network training with attention similarity module; 2) SHNU2: 10-way 5-shot prototypical network training with attention similarity module; 3) SHNU3: 10-way 5-shot ResNet model with embedding propagation; All the systems are with SpecAugment [2] and inference-time [3] data augmentation techniques to enhance the model robustness.

## 2. DATA

In this challenge, the development set provided by the official organizers is pre-split into training and validation sets. The training set consists of four different sub-folders (BV, HV, JD, MT), each for one source class. Along with the audio files multi-class annotations are provided for each. The total duration of whole training set is 14.3 hours, with 10 hours, 3 hours, 10 minutes and 1.16 hours for BV, HV, JD, and MT respectively. The total classes is 19, in which 11 for BV, 3 for HT, 1 for JD and 4 for MT. In addition, the sampling rate is also very different for different sources, it varies from 6kHz (for HT) to 24kHz (for BV).

The validation set comprises of two sub-folders (HV, PB). It includes total 5 hours data, covers 4 classes, 2 for HV with 6kHz sampling rate and 2 for PB with 44.1kHz sampling rate. The two classes for each source are actually the target events and the backgrounds.

## 3. FEATURES

All of our systems use the same Per channel energy normalisation (PCEN) [4] features as used in the official baseline system [5]. They aim to improve the robustness of mel-frequency spectrogram to channel distortion, by combining dynamic range compression (DRC) and adaptive gain control (AGC) with temporal integration. The PCEN is conducted on mel frequency spectrogram and used as input feature. Raw audio is scaled to the range $[-2^{31}, 2^{31} - 1]$ before mel transformation. Detail parameters can be found in [5].

## 4. PROTOTYPICAL NETWORK

Prototypical networks were introduced by Snell et. al in [6]. The core idea of the methodology is to learn an embedding space where points cluster around a single prototype representation of each class. A non-linear mapping from the input space to embedding space is learnt using a convolutional neural network. Class prototype is calculated by taking a mean of its support set in the embedding space. Classification of a query point is conducted by finding the nearest class prototype.

The prototypical networks that used in our `SHNU1` and `SHNU2` systems use the same 4-layer convolutional neural network as the official baseline [5]. All the prototypical networks adopt an episodic training procedure where in each episode, a mini-batch is sampled from the dataset ensuring that each class has an equal representation, post which a subset of the mini batch is used as the support set to train the model and the remaining data is used as query set. The intention of episodic training is to replicate a few-shot learning task.

As the official baseline, we also extract equal length patches from the annotated segments, where each patch inherits the label of its corresponding annotation. And because of the training set is heavily imbalanced in terms of class distribution, both the support and query sets are balanced using oversampling.

## 5. ATTENTION SIMILARITY

As our systems are trained and test on the dataset with segment-level audios (0.2 seconds per segment in our systems), there are some segments include target events shorter than $0.2$s, the segment-level model training strategy may make a model overlook short or transient sound event, in our system `SHNU1` and `SHNU2`, we integrate an attention similarity module in the prototypical network to automatically guide the model pay attention to specific parts of a long audio segment for recognizing relatively short or transient sound events.

The attention similarity module was first proposed in [7], the similarity function can be expressed as follows:

$$\begin{aligned} f_{att\_sim}(\mathbf{X}_q, \mathbf{X}_j) &= \mathbf{A}_q^T(\mathbf{X}_q^T\mathbf{X}_j)\mathbf{A}_j \\ &= (\mathbf{X}_q\mathbf{A}_q)^T(\mathbf{X}_j\mathbf{A}_j) \end{aligned} \tag{1}$$

where $\mathbf{X}_q$ and $\mathbf{X}_j$ are two inputs feature maps of the output (a.k.a. a feature map) from the last convolutional layer of our 4-layer CNN prototypical network. $\mathbf{A}_q$ and $\mathbf{A}_j$ are the attention vectors that computed using another stack of convolutional layers $f_{att}(\cdot)$ by feeding $\mathbf{X}_q$ and $\mathbf{X}_j$ to find the important parts.

This function can be interpreted as we compute the similarity score by using the inner product between two attentional vector $\mathbf{X}_q\mathbf{A}_q$ and $\mathbf{X}_j\mathbf{A}_j$. It allows us to replace the inner product with common distance functions to measure the distance between two attentional vectors. Therefore, in our attention similarity based systems `SHNU1` and `SHNU2`, we replace the simple Euclidean distance that used in the official baseline system with the attention Euclidean distance similarity as in Eq.(1). More details of the attention similarity can be found in work [7].

## 6. EMBEDDING PROPAGATION

The embedding propagation (EP) is an unsupervised non-parametric regularizer for manifold smoothing in few-shot classification. It was first proposed in [8] to improve the model generalization ability to unseen classes. EP outputs a set of interpolations from the network output features using their similarity in a graph. This graph is constructed with pairwise similarities of the features using the radial basis function (RBF). It leverages embedding interpolations to capture higher order feature interactions.

Specifically, given a set of feature vectors of an episode, the pairwise Euclidean distance is computed on each pair of features. Then, these distances are used to calculate an adjacency matrix. After performing the Laplacian of the adjacency matrix, we can obtain a propagator matrix to project each feature vector to another feature space. These propagated features are then can be viewed as a weighted sum of their neighbors, which makes the embedding propagation has the effect of removing undesired noise from the feature vectors. Since this EP operation is simple to implement and compatible with a wide range of feature extractors and classifier, therefore, in our `SHNU3` system, we also regularize our feature embeddings for the training (outputs of 12-layer ResNet of the prototypical network) using the EP to improve the model generalization ability.

## 7. EXPERIMENTS

All our experiments are performed on the development dataset of DCASE 2021 Task5 Challenge. Except pre-trained method, only the official released data as described in section 2 is used to train the networks. Follow the challenge ruler, all the systems are trained as $N$-way 5-shot task. All of the submitted systems are examined on the validation set. Results are show in Table 1.

Table 1: Overall results of our submissions on the validation set of DCASE 2021 Task5.

| ID | Method | F-measure | Precision | Recall |
|---|---|---|---|---|
| Baseline | 10-way, CNN | 41.48% | 32.20% | 58.27% |
| SHNU1 | 3-way, CNN, Attention | 54.69% | 60.99% | 49.57% |
| SHNU2 | 10-way, CNN, Attention | 51.69% | 57.42% | 47.00% |
| SHNU3 | 10-way, ResNet, EP, Pre | 51.39% | 55.88% | 47.56% |

In Table 1, the “CNN, Attention” represents the model structure is the prototypical network with attention. “ResNet” represents use 12-layer ResNet [8] replace the 4-layer CNN in prototypical network. “EP” means using the embedding propagation as a regularized method. “Pre” means using the 17.68 hours strong labeled animal AudioSet [9] as pre-trained dataset.

We train about 16K episodes for 3-way strategy and 4.8K episodes for 10-way strategy, others follow the baseline [5]. Besides, all our systems use the same techniques as follows:

1) SpecAugment and inference-time data augmentation;
2) 5 predictions’ average (re-run 5 times, then average);
3) for post processing, peak picking and median filter with $1/3$ of first five positive events average length as window length.

## 8. CONCLUSION

This technical report presents all the methods that used in our submissions of DCASE 2021 Task5. Experiments on the validation set show that the introduced attention similarity and embedding propagation can improve the performance more than absolute 10% F-measure over the baseline. Moreover, we see that different training strategy and model structure can also affect the system performances.

## 9. REFERENCES

[1] http://dcase.community/challenge2021/.

[2] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech 2019*, 2019, pp. 2613–2617.

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

[4] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, “Trainable frontend for robust and far-field keyword spotting,” in *Proc. ICASSP 2017*, 2017, pp. 5670–5674.

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

[6] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017.

[7] S.-Y. Chou, K.-H. Cheng, J. Jang, and Y.-H. Yang, “Learning to match transient sound events using attentional similarity for few-shot sound recognition,” *ICASSP 2019*, pp. 26–30, 2019.

[8] P. Rodr’iguez, I. H. Laradji, A. Drouin, and A. Lacoste, “Embedding propagation: Smoother manifold for few-shot classification,” in *ECCV*, 2020.

[9] https://research.google.com/audioset/download_strong.html.