FEW-SHOT BIO-ACOUSTIC EVENT DETECTION BASED ON TRANSDUCTIVE LEARNING AND ADAPTED CENTRAL DIFFERENCE CONVOLUTION

Technical Report

Qisheng Huang, Yanxiong Li, Wenchang Cao, Hao Chen

School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

839508665@qq.com, eeyxli@scut.edu.cn

ABSTRACT

In this technical report, we present our submitted system for DCASE2022 Task5: few-shot bio-acoustic event detection. Our system employs the transductive learning strategy, data augmentation and an adapted version of central difference convolution (CDC). Evaluated on the validation set, our method achieves the overall F-measure score of 41.1%.

Index Terms— Few-shot learning, sound event detection, transductive learning.

1. INTRODUCTION

Bio-acoustic event detection is a task to detect whether a certain animal vocalization happens in a given audio clip and when. For this particular task, data scarcity is a very tricky issue. On the one hand, the vocalizations of some species are difficult to collect. On the other hand, the collected data can only be annotated by people with expert knowledge. Therefore, the popular methods based on standard supervised learning that achieves good results on other types of sound event detection may not perform well on this task. In contrast, few-shot learning describes tasks in which an algorithm must make predictions given only a few instances of each class. This paradigm is a perfect fit for the issue mentioned above. It is safe to say that few -shot bio-acoustic event detection is worth exploring as it satisfies the need of users.

In this challenge, the few-shot task is defined as 5-shot problem. Only five annotated calls are provided for the recordings in the official evaluation set. Each recording has a single class of interest which the participants will then need to detect through the recording utilizing only the annotated instances.

2. FEW-SHOT SETTING

Let x and y denote an instance and its ground-truth label respectively. The training and test datasets are $D_S=\{(x_i,y_i)\}_{i=1}^{N_s}$ and $D_q=\{(x_i,y_i)\}_{i=1}^{Nq}$ respectively, where $y_i \in C_t$ for some set of classes C_t. In the few-shot setting, training and test datasets are referred to as support and query datasets, respectively. The number of ways (classes) is $|C_t|$. The number of shots (instances) is marked as N_s and is small in each support set. The goal is to learn a function F to exploit the training set D_S to predict the label of a test instance in D_q.

3. PROPOSED METHOD

In this section, we introduce the methods used in our system, including data augmentation, transductive learning and our proposed adapted central difference convolution.

3.1. Data augmentation

Data augmentation is a commonly used strategy in both computer vision and audio processing tasks to enhance the robustness of target model. The SpecAugment [1] is a simple data augmentation method that is applied directly on the spectrogram. Specifically, the operations of this method include time warping, frequency masking, and time masking.

3.1.1. Time Warping

Given a log mel spectrogram with τ time steps, a random point along the time axis passing through the center of the spectrogram within the time steps $(W, \tau - W)$ is warped either to the left or right by a distance ω chosen from a uniform distribution from 0 to the time warp parameter W.

3.1.2. Frequency masking

Masking is applied on *f* consecutive mel frequency channels $[f_0, f_0+f)$ where *f* is first chosen from a uniform distribution from 0 to the frequency mask parameter F and f_0 is chosen from $[0，v-f)$. v is the number of mel frequency channels.

3.1.3. Time masking

Masking is applied on *t* consecutive steps $[t_0, t_0+t)$, where t is first chosen from a uniform distribution from 0 to the time mask parameter T and t_0 is chosen from [0, *τ-t*). This operation is applied during the training phase, which is introduced in the following subsection.

3.2. Transductive learning

For a given few-shot task, we have a support set $S=\{X_S,Y_S\}$ and a query set $Q=\{X_Q,Y_Q\}$, where Y_Q denotes the labels to be predicted. Let $f_\theta : X \rightarrow Z \in R^d$ denote the feature extractor and Z represents the set of extracted features. In addition, a base dataset $D_{base}=\{X_{base,}Y_{base}\}$ is provided to pre-train the feature extractor.

The idea of transductive learning is to leverage the unlabeled instances in the query set to facilitate the network update on the labeled instances in the support set. Our method is inspired by the work in [2] in which the mutual information (MI) between the query instances and their predicted labels during inference in maximized. This method is referred to as **T**ransductive **I**nformation **M**aximization (TIM). Specifically, we first train a backbone network f_θ on the training set provided by the organizer following the standard supervised learning paradigm. We compute the cross-entropy loss between the predicted label and the ground-truth label and update the model parameters through the algorithm of back-propagation. Next, the backbone model is frozen and is used as the feature extractor, and we construct a classifier parameterized by $W=[w_1,...,w_K]\in R^{K\times d}$ and initialize its parameters with the prototypes [3] of each class in the support set. The posterior distribution over labels given features is given by $p_{ik}=P(Y=k|X=x_i;W,\theta)$ and similarly the marginal distribution over query labels is $\hat{p}_k=P(Y=k;W,\theta)$. p_{ik} and $\hat{p}_k$ are calculated by

$$p_{ik}=\frac{\exp(w_k\cdot z_i)}{\sum_{c=1}^{K}\exp(w_c\cdot z_i)},\widehat{p}_k=\frac{1}{Q}\sum_{i\in Q}p_{ik}\ . \tag{1}$$

Finally, we finetune the classifier using the loss function defined by

$$L_w=\lambda_{CE}\cdot CE-I(X_Q;Y_Q) \tag{2}$$

where λ_{CE} is a hyper-parameter, $I(X_Q;Y_Q)$ is the mutual information. The definition of CE and $I(X_Q;Y_Q)$ are given by

$$I(X_Q;Y_Q)=-\sum_{k-1}^{K}\widehat{p}_k\log\widehat{p}_k+\frac{1}{|Q|}\sum_{i\in Q}\sum_{k=1}^{K}p_{ik}\log p_{ik} \tag{3}$$

and

$$CE=-\frac{1}{|S|}\sum_{i\in S}\sum_{i=1}^{K}y_{ik}\log(p_{ik})\ . \tag{4}$$

The overall evaluation process is illustrated in Fig. 1. In our experiment, we set the hyper-parameter λ_{CE} as 0.1.

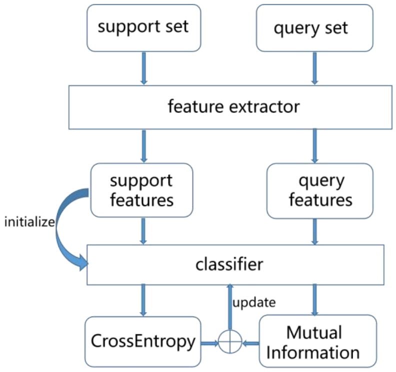

Fig. 1 The overall process of transductive learning.

3.3. Central difference convolution

Central difference convolution (CDC) is proposed in [4] for robust feature representation. For the sake of brevity, in this subsection the convolutions are described in 2D.

3.3.1. Vanilla Convolution

We denote the basic 2D spatial convolution in convolutional neural network as vanilla convolution. The operation includes two main steps: sampling local receptive field region R over the input feature map x, and aggregation of sampled values via weighted summation. Hence, the output feature map y can be formulated as

$$y(p_0)=\sum_{p_n\in R}w(p_n)\cdot x(p_0+p_n) \tag{5}$$

where p_0 denotes current location on both input and output feature maps while p_n enumerates the locations in R.

3.3.2. Central Difference Convolution

The CDC is inspired by the famous local binary pattern [4, 5], which describes local relations in binary central difference way. By combining central difference with vanilla convolution, the CDC is defined by

$$y(p_0)=\alpha\cdot\sum_{p_n\in R}w(p_n)\cdot(x(p_0+p_n)-x(p_0))$$

$$+(1-\alpha)\cdot\sum_{p_n\in R}w(p_n)\cdot x(p_0+p_n) \tag{6}$$

where hyper-parameter $\alpha\in[0,1]$ balances the contribution between intensity-level and gradient-level information. In our experiment, α is set to 0.7.

3.3.3. Implementation of CDC

In order to efficiently implement central difference convolution in deep learning framework. Eq. 6 can be decomposed into

$$y(p_0)=\sum_{p_n\in R}w(p_n)\cdot x(p_0+p_n)+\alpha\cdot(-x(p_0)\cdot\sum_{p_n\in R}w(p_n)) \tag{7}$$

3.3.4. Adapted Central Difference Convolution (ACDC)

As image features are quite different from the audio counterpart, we adopt the CDC mentioned above for audio tasks. The adopted version is given by

$$y(p_0)=\sum_{p_n\in R}w(p_n)\cdot x(p_0+p_n)-\alpha\cdot(-x(p_0)\cdot\sum_{p_n\in R}w(p_n)) \tag{8}$$

In our experiment, we directly replace all the vanilla convolution kernel with ACDC without making any other changes to the network architecture.

4. EXPERIMENT

4.1. Experiment setups

The dataset is from DCASE2022 task5, including development and evaluation sets. The development set is pre-split into training and validation sets. For all the experiments, we employ the even-based F-measure, which is the same as the official baseline [6]. Our system directly uses the feature extraction module provided by the official baseline [6] with the same parameters except for setting a fixed segment length and hop length of 17 frames and 4 frames respectively. The network in [3] is used as the backbone of our network, which consists of 4 convolutional layers. The difference between our network and the network in [3] is that we add a dense layer after the last convolution layer to compute the classification results during training.

4.2. Ablation study

In his part, we study the influence of each component of our methods, and the results are presented in Table 1.

Table 1. Influence of each component of our method.

Method	F-score (%)
TIM	43.646
TIM+SPEC	43.974
TIM+ACDC	52.77
TIM+SPEC+ACDC	54.59

4.3. Experiment result

Table 2 shows the experiment results on the validation set. Our method achieves F-score of 41.09%, precision of 51.08% and recall of 34.38% on the development set.

Table 2. Experiment results on the validation set.

Method	Precision (%)	Recall (%)	F-score (%)
Baseline [6]	36.34	24.96	29.59
Ours	60.88	49.48	54.59

5. CONCLUSIONS

In this technical report, we develop a few-shot bio-acoustic event detection system based on transductive learning and adapted central difference convolution. In addition, we also employ some data augmentation method to improve the system performance. However, much work still remains to be done to how different parts of the system influence the performance.

6. REFERENCE

[1] 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, pp. 2613-2617.

[2] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Pia-ntanida, and I. B. Ayed, “Information maximization for few-shot learning,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[3] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4077-4087.

[4] Z. Yu, C. Zhao, and Z. Wang, “Searching central difference convolutional networks for face anti-spoofing” in *CVPR*, 2020, pp. 5295-5305

[5] Z. Boulkenafet, J. Komulainen, and A. Hadid. “Face anti-spoofing based on color texture analysis,” *IEEE ICIP*, pp. 2636-2640, 2015.

[6] https://github.com/c4dm/dcase-few-shot-bio-acoustic/tree/ma-in/baselines.