FEW-SHOT BIOACOUSTIC EVENT DETECTION USING BEATS

Technical Report

Femke Gelderblom[1*], *Benjamin Cretois*[2*], *Pål Johnsen*[3], *Filippo Remonato*[3], *Tor Arne Reinen*[1]

[1] Acoustics, SINTEF Digital, Trondheim, Norway
[2] Environmental Data, Norwegian Institute for Nature Research, Trondheim, Norway
[3] Mathematics and Cybernetics, SINTEF Digital, Trondheim, Norway

ABSTRACT

Our method for the DCASE Challenge 2023 combines BEATs with Prototypical Networks. BEATs, standing for Bidirectional Encoder representation from Audio Transformers, is a newly-released architecture by Microsoft for audio tokenisation and classification. BEATs combines a tokenizer and a semi-supervised audio classifier which learn from each other to improve the classification of audio samples. Prototypical Networks, instead, can be briefly described as a neural network-based clustering algorithm. Somewhat resembling a K-means clustering, Prototypical Networks classify samples based on their distance from the classes' prototypes (what would be the centroids in a K-means setting). Since the prototypes are constructed from a small set of examples from each class, called the support set, Prototypical Networks are well suited to handle few-shot learning settings like the DCASE Challenge. In our method, we combine the two by using BEATs as a feature extractor, constructing informative features which are used by the Prototypical Network to perform the prototypes' construction and subsequent classification of test audio samples. We obtain a F1 score of 0.36 on the validation dataset.

***Index Terms*—** DCASE, Few-shot Learning, Prototypical Network, Acoustic tokenizers

1. INTRODUCTION

Few shot classification is a task in which a classifier must be adapted to accommodate new classes not seen in training, given only a few examples of each of these classes [1]. Classifying unseen objects given very few examples is trivial for humans, who have the ability to perform one-shot classification (i.e. only a single example of each new class is given) with a high degree of accuracy [1]. However, few-shot classification has proven to be challenging for even state-of-the-art machine learning algorithms.

In this technical report we describe our method to tackle the task 5 of the Detection and Classification of Acoustic Scenes and Events challenge 2023 (DCASE2023): Few-shot bioacoustic event detection.

2. DATASET

To train, validate and test our algorithm we used the dataset provided by the Detection and Classification of Acoustic Scenes and Events (DCASE 2023) challenge. The training set consists of four different sub-folders deriving from a different source each (BV, HT, JD, MT, WMW, see [2] for more information about the dataset). Along with the audio files, multi-class annotations are provided for each. The total duration of whole training set is 14.3 hours, divided as 10 hours, 3 hours, 10 minutes, and 1.16 hours for BV, HV, JD, and MT respectively. The total number of classes is 19, of 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, varying from 6 kHz for HT, to 24 kHz for BV. The validation set comprises of two sub-folders (HV, PB). It includes a total of 5 hours data covering 4 classes: 2 for HV with 6 kHz sampling rate and 2 for PB with 44.1 kHz sampling rate. The two classes for each source are actually the target events and the backgrounds.

3. METHOD

Central to our pipeline is the prototypical network as described in [3]. Specific for our approach are our preprocessing steps and the fact that we rely on the BEATs model to encode our input samples into the latent space where distances are calculated. Details of these steps are given in the next two subsections. Following these, we provide the differences between the submitted systems.

3.1. Preprocessing

The preprocessing pipeline is summarised in Figure 1 and consists of the following steps:

1. Load the raw audio waveform contained in the wav file
2. Resample to 16 kHz
3. Normalize
4. Denoise
5. Obtain Mel-filter bank features

The denoising method of step 4 relies on a method called "spectral gating" as described in [4]. For our system, we relied on the implementation provided in [5]. Background noise was assumed to be stationary.

The dataset provided by the challenge contains many audio events from different types of sources, which therefore also vary a lot in duration: some animal vocalisations are very short, others are much longer. This duration is also affected by how the human annotator has labelled the audio events. For example, in some cases each 'chirp' from a bird has been tagged separately, while in other cases longer episodes containing repeated bird calls are tagged as a single event.

[*] These authors contributed equally to this work

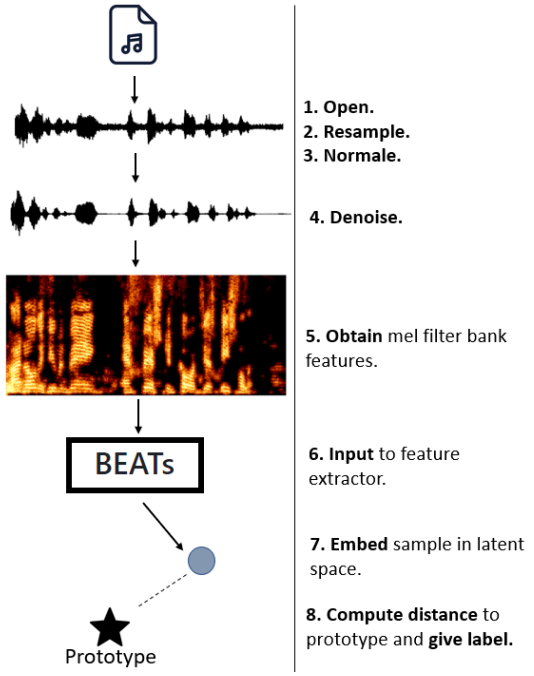

Figure 1: Summary figure representing the pipeline of our system for a query sample. First, we open the raw audiofile, resample to 16kHz and normalise the audio (steps 1,2,3). We then denoise the audio using [4] (Step 4) and convert the audio into a mel spectrogram (Step 5). The mel-spectrogram is passed through BEATs to create an embedding (Step 7) and the distance to all class prototypes is computed. The embedding is given the same label as the closest prototype (Step 8).

It is crucial that the Mel-filter bank features obtained at step 5 are always of equal dimensions, so that they can be further encoded by the BEATs network. At the same time, it is important to provide the BEATs network with all the information relevant to the classification of the audio event. To compensate for the variation in duration of the audio events in the dataset, the frame shift used to calculate the Mel-filter bank features was varied from 1 ms to 8 ms. The frame shift in ms was calculated from the 5 support samples as follows:

$$\text{frame shift} = \max\left(\left[\frac{\min(l,1)}{128}1000\right],1\right), \quad (1)$$

where l is the shortest duration of the support samples, which varies for each file and class. Here, 128 was the number of frame shifts contained in the feature provided to the BEATs model. As such, this feature is minimally 128 ms long (for the shortest events of the dataset) and maximally 1 s long (for the longest events of the dataset).

3.2. Presentation of the BEATs model

Our method builds on the newly-released BEATs model [6] by Microsoft. BEATs is a powerful architecture for audio classification composed of two main blocks: A tokenizer, which learns to split the audio signal into relevant semantic segments, i.e. segments where the signal of interest is active vs inactive; and a classifier which takes the audio tokens and assigns a human-readable label like 'cat', 'dog', 'car', etc. We will not include the details of BEATs here, among other reasons because the complexity of the architecture and the presentation in [6] make it difficult to truly understand them within the limited timeframe offered by the DCASE challenge, but we hope the interested reader will find the original publication elucidating. Regardless, the architecture produces clearly good results: Figure 2 shows an example test result we obtained while learning how to use BEATs, applying it to the ECS50 dataset [7] as a feature-extractor. By "feature extraction" (Step 6 Figure 1) we mean we used BEATs' activations in the second-to-last layer, i.e. right before the final classification step. BEATs comes pre-trained on the AS-2M dataset, and in Figure 2a we show the clustering following the feature extraction on 10 classes from the ESC50 dataset without any additional fine-tuning. In Figure 2b instead, BEATs has been fine-tuned on the 10 classes we chose. It is worth noting that we trained for only 7 epochs during the fine-tuning stage.

3.3. Prototypical network using BEATs

Even though BEATs is a powerful architecture for audio classification, it is necessary to use it in tandem with other methods for few-shot classification. Our system uses a prototypical network using BEATs as feature extractor in tandem with episodic training for better domain generalisation [8]. Episodic training is a method where a model is trained by simulating few-shot learning scenarios through episodes (i.e. a simulated scenario where a model is presented with a small set of labeled examples from different classes), allowing it to learn from a small number of examples per class.

More specifically, to train the models, for each episode we use five classes (i.e. number of ways = 5) we build the class prototypes using five embedded samples (i.e. number of support = 5) and compute the loss using the distance of 10 query samples (i.e. number of query = 10). For model validation and evaluation we used number of ways = 2 (i.e. for POS and NEG classes), number of support = 2 and number of query = 3. We trained the models for 100 episodes per epoch using an early stopping strategy. After the sample has been projected into the latent space (Step 7 Figure 1), we compute the distance between embedded sample and the prototype we use euclidean distance as recommended in [3].

We tested our pipeline on a subset of ECS50 comprising only five classes with 15 samples each. Results are displayed in Figure 2c where we obtained an accuracy of over 90%. We also clearly see that the samples are in close vicinity to the class prototypes.

3.4. Submitted System Details

The submitted systems are all similar, but with small variations.

- System 1: The system where all preprocessing steps are included, but with only 50% overlap in the features that are presented to the system
- System 2: Equal to system 2, but with the added step that the 20 events with the least distance to the prototype are added as additional supports.

4. CONCLUSION

This technical paper describes the pipeline for our submission to DCASE task 5: Few-shot bioacoustic event detection. The pipeline

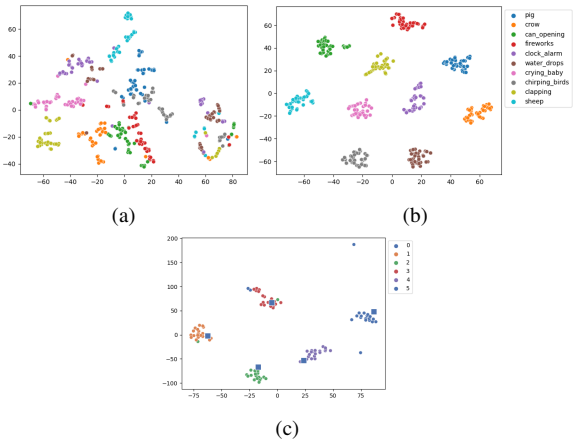

Figure 2: A simple test of feature extraction and clustering on the ECS50 dataset, using (a): "raw" BEATs on 10 classes; (b): Fine-tuned BEATs; (c) BEATs and prototypical network on 5 classes. BEATs latent space contains 768 dimensions and we used t-sne to represent the embeddings in a two dimensional space.

is based on a prototypical network with Microsoft's recently released 'BEATs' model as encoder. Results and analysis of its performance will be included in a later publication.

5. REFERENCES

[1] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, no. 33, 2011.

[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] 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., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf

[4] T. Sainburg, M. Thielk, and T. Q. Gentner, "Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires," *PLoS computational biology*, vol. 16, no. 10, p. e1008228, 2020.

[5] T. Sainburg, "timsainb/noisereduce: v1.0," June 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3243139

[6] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," 2022.

[7] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[8] 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.