TRANSFORMER-BASED BIOACOUSTIC SOUND EVENT DETECTION ON FEW-SHOT LEARNING TASKS

Liwen You, Erika Pelaez Coyotl, Suren Gunturu, Maarten Van Segbroeck

Amazon.com

ABSTRACT

Automatic detection of bioacoustic sound events is crucial to monitor wildlife. With a tedious annotation process, limited labeled events and large volume of recordings, few-shot learning (FSL) is suitable for such event detections based on a few examples. Typical FSL frameworks for sound detection make use of Convolutional Neural Networks (CNNs) to extract features. However, CNNs fail to capture long-range relationships and global context in audio data. We present an approach that combines the audio spectrogram transformer (AST), a data augmentation regime and transductive inference to detect sound events on the DCASE2022 (Task 5) dataset. Our results show that the AST model performs better on all recordings when compared to a CNN based model. With transductive inference on FSL tasks, our approach has 6% improvement over the baseline AST feature extraction pipeline. Our approach generalizes well over sound events from different animal species, recordings and durations, suggesting its effectiveness for FSL tasks.

Index Terms— Audio spectrogram transformers, sound event detection, few-shot learning, transductive inference

1. INTRODUCTION

Automatic detection of bioacoustic sound events is crucial to monitor wildlife over long periods of time in a scalable and minimally invasive way. Per-species sound event detection requires identification, classification and quantification of vocalizations as individual acoustic events, which is further complicated by the quality of the recording, the presence of background noises and the occurrence of multiple sound events from different sources (species) at the same time. Furthermore, the amount of labeled data is minimal since monitor devices produce a great amount of recordings that would need hundreds of hours of manual effort to be properly labeled (e.g., determine time marker, species and/or call-type). Few-shot learning is a highly promising paradigm for sound event detection that aims to overcome these challenges.

A few-shot learning task is defined by the amount of labeled data (support set) and unlabeled data (query set) in the task. Learning is achieved from a few examples in a support set and prediction is performed on query set. In our FSL setup, we only use the first five positive events in a recording as a support set, while we predict the remaining recording in the query set. We treat all events between two positive labeled events as negative events. Because of the size of the support set, generating good feature representations of sounds is a crucial part of the pipeline to get good prediction performance on the query set. With this in mind, we apply transfer learning from a pretrained model on audio spectrograms to a subset of our labeled recordings with the aim to classify the different event classes. We then use this model to extract sound feature embeddings to build our FSL models. Another challenge we are facing in the sound event detection task, is the presence of a wide range of event durations, which makes the feature extraction difficult to generalize well on events of varying duration ranges. Besides accurate classification of sounds events, we also need to detect the event start and end time within sufficiently prediction margins.

2. DATA AND TASK DESCRIPTION

The dataset used in our study is based on the few-shot bioacoustic event detection dataset used for the DCASE 2022 task-5 challenge [1]. This dataset contains one training set and one test set. The training set consists of 174 recordings across different bioacoustic sounds; and the test set contains 18 recordings of 3 different subsets, named as HB, PB and ME. Each recording is associated with additional meta data, which contains sound event start and end time, together with the class names. Different recordings or sets have different event durations. The median event duration for the 3 sets are 0.02-0.17ms for PB, 1.3-19.35ms for HB and 0.14-0.24ms for ME. There are three recordings with a median duration of only 0.02ms in the PB set. Our approach has limitations for events of extreme short duration and not to bias the result by those we have excluded these recordings. We discuss this in the conclusion section for future work.

3. METHOD

We transform the recordings to the frequency domain by converting wave data into Mel scale time-spectral representations. To detect the events of interest, we extract small segments of length L, along the time axis of the spectrogram. We then predict the probability for each segment of being positive, i.e. containing a partial or whole positive event for the target sound class. After that, the predicted positive segments are grouped together into a positive event upon which the event start and end times are determined.

We use recordings from the training dataset to first train a feature extractor. We then apply the feature extractor on the FSL tasks to generate embedding vectors that will serve as input to our FSL classifier. For recordings in the training dataset, we generate positive and negative segments along the whole recordings and label them with the corresponding sound event classes. We treat all events happening between two consecutive positive events as negative events and assign them the background class label.

Since individual FSL tasks might have their own specific features which are not captured by the general feature extractor, we fine-tune the feature extractor by just using the support set of an FSL task as a binary classifier to further refine the feature representation with transductive inference on individual FSL task. Due to very few positive data points for events with short duration, we use data augmentation to increase the positive training samples with frequency

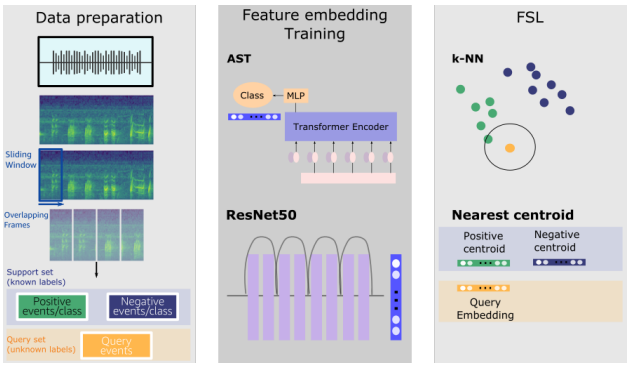

Fig. 1. Overall architecture (extractor and classifier) and data preparation pipeline.

and time masking on spectrum space. Our overall architecture and data pipeline are illustrated on Figure 1.

3.1. Feature extraction

We use two different architectures to generate embeddings for the FSL task. The first one is ResNet, a convolutional neural network (CNN) architecture, and the second is AST, a transformer-based architecture for spectrograms. ResNet has shown great success on image classification problems since it was first introduced [2]. We use the ResNet configuration with three-layer ResNet blocks, followed by an adaptive average pooling layer and a fully connected layer to predict segment classes. ResNet and other CNN architectures have been commonly used to create audio models. In this work, we use both non-pretrained and pretrained ResNet [3], trained on AudioSet [4]. We benchmark the AST architecture against ResNet for the sound event detection problem in this paper.

The Audio Spectrogram Transformer (AST) was proposed for audio classification, leading to state of the art results in several datasets [5]. AST is the first convolution-free, purely attention-based method for audio classification. A 2D audio spectrogram is split into a sequence of 16x16 patches with overlap, and then linearly projected into a sequence of 1-D patch embeddings. Each patch embedding is added with a learnable positional embedding. We use a pre-trained AST model, trained on AudioSet set, and further adapt the model to serve as the feature extractor module by fine tuning it on the generated audio segment dataset from our training recordings. The pre-trained AST model has been trained on spectrogram extracted using a sample rate of 16kHz.

3.2. Transductive inference

For individual FSL task, we further improve feature extraction by fine tuning the AST model as a binary classifier on its support set, so that specific features for that FSL task could be captured by this second fine tuning process. Due to the data scarcity for events with short durations, we use data augmentation technique with frequency and time masking [6] to increase our training samples. Since we have more negative samples, we use a balanced data set as our training dataset. During the fine tuning process, we build a binary classifier for individual FSL task. However, we do not use the predictions from this model on final classification task. Instead, we use it as a feature extractor. Our experiments show better results with this approach.

Event duration	0-20	20-60	60-100	>100
Segment Length	n	20	60	128
Hop length	n/4	5	15	32

Table 1. Dynamic segment and hop length on FSL tasks. Event duration is counted in number of frames. If the event duration is less than 20 frames, we use the original event duration as segment length. Hop length is 1/4 of the segment length and minimal as 1 frame.

3.3. Classifier

When detecting events on few-shot learning tasks, we treat each task individually and use a metric-based few-shot learning approach. Each segment is projected into an embedding space by applying the feature extraction model. We generate positive/negative prototype embeddings by averaging all positive/negative segments from the support set. We use the following equation to get positive segment probability:

$$P(positive)_i = \frac{\exp\left[f(w_p, s_i)/T\right]}{\exp\left[f(w_p, s_i)/T\right] + \exp\left[f(w_n, s_i)/T\right]},$$

where f is the Euclidean distance metric; w_p is the positive segment prototype embedding; w_n is the one for negative segments prototype embedding; s_i is the ith segment and T is the temperature. We compare the above metric based classifier with a K-nearest neighbor (KNN) method (K=1) to predict positive segments. Due to random positive samples generated with data augmentation, we use an ensemble as our final classifier to combine seven classifiers built with different sets of augmented data.

3.4. Data processing

All recordings are down-sampled to 16kHz and STFT (Short Time Frequency Transform) is applied with a 10ms frame shift and a 25ms frame length to generate 128-dimensional Mel-Spectrograms. Each Mel-Spectrogram is normalized at the recording level. We extract small segments (128 frames) from events with specified hop length (shifting window with 32 frames) to generate our training dataset for feature embeddings. If a segment is shorter than 128 frames, it will be padded by repeating the segment information until we reach the desired number of frames.

For the tasks in the test set, we have a wide range of event durations. When using large segment length, extracting event features and their corresponding start and end time for short events is hard. Therefore, we use a dynamic segment length to extract segments. First, we get a medium event duration on the first 5 positive events and then select final segment and hop length for that recording, according to Table 1. This dynamic segment setup could be further tuned for the best performance.

3.5. Post processing

After predicting the probability of a segment containing the target sound, a filter is applied to remove segments with low probabilities using a threshold of 0.5. Thereafter, we merge adjacent segments to an event. If a predicted event duration is less than 60% of the minimal duration on the first five positive events in the support set, it is removed from the final predicted event list. The predicted events contain start and end time, which is used to match ground truth events in the query set for each recording. According to the

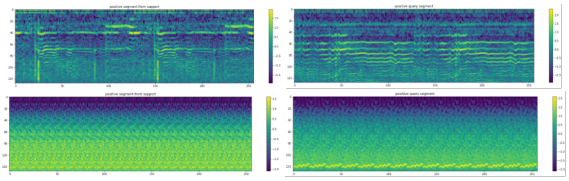

Fig. 2. Spectrogram comparison between support and query sets from short and long events.

DCASE challenge, the Intersection Over Union (IOU) value of 0.3, is used to define event match. Further details can be found on the DCASE homepage (https://dcase.community/).

4. EXPERIMENT

4.1. Comparing AST and ResNet architecture on FSL tasks

We first run our experiments by applying different configurations based on feature extractor and classifier used during FSL tasks without transductive inference. We compare our results on harmonic metrics (precision, recall and F1-score) of the three sets for different combinations of ResNet/AST, prototype/KNN classifier and segment lengths (128, 256, 512 frames). For the pretrained ResNet model, we only run it with 256 segment lenght setup and compare it with the corresponding non-pretrained ResNet. We choose the hop length as one fourth of the segment length. Our results are shown in Tables 2, 3, 4 and 5.

As can be seen from the tables, the AST model performs better across all the three sets, obtaining the best precision score of 70.31, a recall of 69.55 and F1-score of 68.81. This shows that the feature embeddings learned by the AST model are generalizing better on events with different acoustic characteristics and durations. The models with segment length of 256, work better on our dataset. Models based on ResNet show good results on events within the HB and ME sets, but not on the PB set, even with pretrained ResNet. One explanation might be that the ResNet model does not capture location based information well [5]. On short sound events, the sound frequency might appear on specific narrow time-frequency patches. Tuning the CNN kernel size, could improve these results and we plan to investigate this further in future work. When comparing classifiers, it appears that the KNN (K=1) generalizes better than prototype based approach on most of the FSL tasks.

4.2. Effect of data augmentation and transductive inference

We fine tune the AST model to further improve feature extractor for individual FSL task without changing other setups. Our results show consistent improves on HB and ME sets across the three metrics. The overall F1-score based on the three sets increases from 68.8 to 73.1 (6% relative improvement) with segment length as 256, comparing to the best result without using transductive inference. The experiment result is in Table 6. Since we do not freeze parameters of different layers of the AST model, our fine tuned model might overfit to the support set of PB set. We will explore partial fine tuning in our future work.

Extractor/Classifier	Seg	HB	ME	PB	All
AST/Prototype	128	94.0	56.0	29.0	47.6
AST/Prototype	256	96.3	51.8	41.6	55.8
AST/Prototype	512	94.4	47.7	12.7	27.3
ResNet/Prototype	128	92.6	24.7	2.2	5.9
ResNet/Prototype	256	95.9	54.7	5.0	13.2
ResNet/Prototype	512	93.7	63.4	4.4	11.8
AST/KNN	128	95.1	70.8	55.5	70.3
AST/KNN	256	95.5	57.5	61.7	**68.1**
AST/KNN	512	95.6	52.6	36.7	52.9
ResNet/KNN	128	90.9	44.0	2.4	6.8
ResNet/KNN	256	92.4	81.3	13.1	30.2
ResNet/KNN	512	93.2	69.5	15.3	33.2

Table 2. Few-shot learning task precision results with different feature extractor and classifiers.

Extractor/Classifier	Seg	HB	ME	PB	All
AST/Prototype	128	78.3	80.8	47.1	64.6
AST/Prototype	256	78.9	82.7	54.4	69.5
AST/Prototype	512	79.5	78.9	48.5	65.4
ResNet/Prototype	128	77.5	84.6	38.3	50.3
ResNet/Prototype	256	77.5	90.4	43.4	63.8
ResNet/Prototype	512	78.3	86.5	37.5	58.8
AST/KNN	128	79.6	65.4	41.2	57.5
AST/KNN	256	80.8	80.8	54.4	**69.6**
AST/KNN	512	82.2	76.9	37.5	57.9
ResNet/KNN	128	84.1	43.1	3.6	16.0
ResNet/KNN	256	77.5	75.0	21.3	41.0
ResNet/KNN	512	78.4	78.9	21.3	41.5

Table 3. Few-shot learning task recall results with different feature extractor and classifiers.

5. VISUALIZATION AND QUALITY ANALYSIS

5.1. Data visualization

We show two examples of Mel-Spectrogram from positive segment events. The first recording is R4-cleaned-recording-17-10-17 from HB set with the longest event duration of 11.51ms in the validation set. The second one is recording BUK1-20181011-001004 from PB set with only 0.16ms event duration. In Figure 2, we show the visual representation of support (left column) and query (right column) samples from the R4-cleaned-recording on the top row. It can be observed that even though the event duration is long, the similarity between positive segments from support set and query set is not very evident. We find a similar case when the events are short in duration as we can see from the Mel-Spectrogram of the second recording.

5.2. Model prediction visualization

To understand how far off from the event start and end are being predicted by our model, we generate plots of the event predictions compared to the annotated data. In Figure 3, the top plot shows the predicted positive segment probabilities across the query set of R4-cleaned-recording-17-10-17. Those predicted positive segments are merged into predicted positive events. As we can see when the match is accurate, the predictions on positive segments are confident. The

Extractor/Classifier	Seg	HB	ME	PB	All
AST/Prototype	128	85.4	66.1	35.9	54.8
AST/Prototype	256	86.7	63.7	47.1	61.9
AST/Prototype	512	86.3	59.4	20.2	38.5
ResNet/Prototype	128	84.4	38.3	4.1	10.6
ResNet/Prototype	256	85.7	68.1	9.0	21.9
ResNet/Prototype	512	85.3	73.2	7.9	19.6
AST/KNN	128	86.7	68.0	47.3	63.3
AST/KNN	256	87.6	67.2	57.8	**68.8**
AST/KNN	512	88.4	62.5	37.1	55.3
ResNet/KNN	128	84.1	43.1	3.6	9.5
ResNet/KNN	256	84.3	78.0	16.3	34.8
ResNet/KNN	512	85.2	73.9	17.9	36.9

Table 4. Few-shot learning task F1-score results with different feature extractor and classifiers.

Classifier	Metric	HB	ME	PB
P-ResNet/KNN	Precision	93.2	74.0	3.4
	Recall	80.7	71.1	5.0
	F1	86.5	72.5	4.1
ResNet/KNN	Precision	92.4	81.3	13.1
	Recall	77.5	75.0	21.3
	F1	84.3	78	16.3

Table 5. Few-shot Learning task precision, recall and F1-score performance of pretrained ResNet (P-ResNet) or non-pretrained ResNet with 256 segment length.

lower two plots are for recording file-97-113, which is difficult to predict. We can see that a big portion of predicted positive segments have low probabilities.

6. RELATION TO PRIOR WORK

Different techniques have been explored to solve the FSL challenge in the DCASE task 5 dataset. Most of them rely on CNNs to extract features. In this work we explore the use of a pre-trained transformer-based model to substitute CNNs for feature extraction as well as data augmentation and transductive inference.

Previous works like [7, 8, 9] show that CNNs are the gold standard for feature extraction in audio FSL use cases. These networks are usually trained from scratch and do not take the advantage of pretrained models on large datasets. We have seen large audio models with high performing sound detection algorithms like HuBERT [10] and sound classification like AST [5].

Once the feature extractor has been trained, several approaches can be taken to determine the label of an incoming recording. The most common method is to use distance based approaches like Nearest centroid and KNN [9]. Our approach combines proven effectiveness of transformer architectures, inspired by the Vision Transformers (VIT) [11], with pretrained model, and careful attention to data preparation and sampling of the few-shot examples to train the models. Furthermore, previous work [12, 13] has shown the effectiveness of transductive inference for FSL tasks. We adopt this approach by incorporating information from the support set to the feature extractor for each recording.

Transductive inference	HB	ME	PB	All
Precision	96.1	81.8	65.0	**78.9**
Recall	81.4	86.5	49.3	**68.0**
F1-score	**88.1**	**84.1**	**56.1**	**73.1**

Table 6. Few-shot learning task performance with transductive inference.

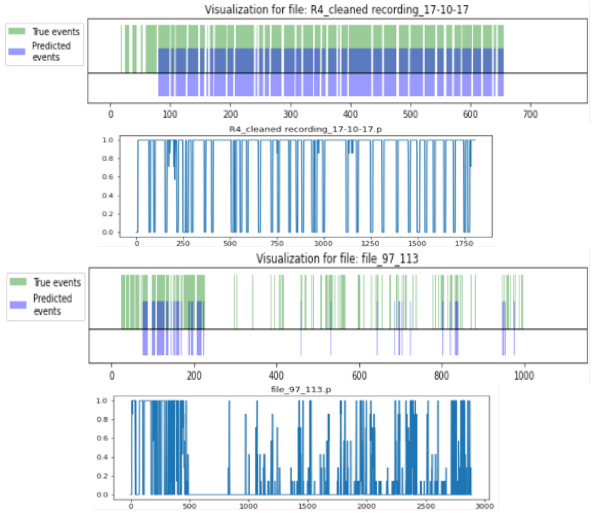

Fig. 3. Model prediction performance visualisation. The first five events are from the support set, so there are no predictions on them.

7. CONCLUSION

Our work shows that the embedding features extracted with the fine-tuned AST model generalize well over different sound events, even on events of short duration. When compared with a CNN-based ResNet model, the AST model shows performance improvement on all FSL tasks. For sound event detection, a dynamic segment length is important to capture features for a wide range of event durations. For the FSL tasks, our results show that KNN works better than a prototype based classifier. Refining feature extraction with transductive inference can further improve model performance.

In this work, we only use Mel-Spectograms. Other spectrum features like PCEN (Per-Channel Energy Normalization), Chirplet Spectrogram, contain different and complementary information. We plan to explore different features and even combine them in future work. Furthermore, semi-supervised technique can be used to leverage unlabeled query data by incorporating different loss functions, i.e. cross entropy, Mutual Information, contrastive loss and Kullback-Leibler Divergence [13].

In our experiments, we exclude the three shortest recordings in the PB set. One reason is that the training data set does not have any recordings shorter than 0.10 ms. In order to solve that, we could explore better dynamic segment schema and patch size; and include short events in our training dataset. Our AST model is based on pretrained models with sample rate 16kHz. If our FSL tasks contain events on high frequency, it affects the feature extraction. We will investigate the sampling rate impact in our future work.

8. REFERENCES

[1] Veronica Morfi, Inês Nolasco, Vincent Lostanlen, Shubhr Singh, Ariana Strandburg-Peshkin, Lisa F. Gill, Hanna Pamuła, David Benvent, and Dan Stowell, "Few-shot bioacoustic event detection: A new task at the dcase 2021 challenge," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, 2021, Detection and Classification of Acoustic Scenes and Events 2021 Workshop, DCASE2021 ; Conference date: 15-11-2021 Through 19-11-2021.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*.

[3] Turab Iqbal Yuxuan Wang Wenwu Wang Qiuqiang Kong, Yin Cao and Mark D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing 28*. IEEE, 2020, pp. 2880–2894.

[4] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017.

[5] Yuan Gong, Yu-An Chung, and James R. Glass, "AST: audio spectrogram transformer," *CoRR*, vol. abs/2104.01778, 2021.

[6] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[7] Jiang jian Xie, Chang qing Ding, Wen bin Li, and Cheng hao Cai, "Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks," 2018.

[8] Dan Stowell, "Computational bioacoustics with deep learning: a review and roadmap," *PeerJ*, vol. 10, pp. e13152, 2022.

[9] Steinar Laenen and Luca Bertinetto, "On episodes, prototypical networks, and few-shot learning," 2020.

[10] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," 2021.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[12] Dongchao Yang, Helin Wang, Yuexian Zou, Zhongjie Ye, and Wenwu Wang, "A mutual learning framework for few-shot sound event detection," 2021.

[13] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?," .