LEARNING IN THE WILD: BIOACOUSTICS FEW SHOT LEARNING WITHOUT USING A TRAINING SET

*Víctor Aguado, Joan Navarro, Ester Vidaña-Vila**

La Salle Campus Barcelona, University Ramon Llull, Barcelona, ES
* Corresponding author: ester.vidana@salle.url.edu

ABSTRACT

Few-shot learning is a machine learning approach in which a pre-trained model is re-trained for new categories with just a few examples. This strategy results very convenient for tasks with a dynamic number of categories as typically happens in acoustic data. The purpose of this paper is to explore the possibility of skipping this pre-training process and using as training data only the five first shots of an audio file together with the silence between them. For the experimental evaluation, data belonging to the Validation set of Task 5 DCASE Challenge 2023 is used, purposely neglecting the Training set. This challenge consists of detecting animal species using only five positive examples. In this exploratory work, three learning methods have been compared: a ResNet architecture with a prototypical loss, a ProtoNet and an XGBoost classifier. In all cases, spectrograms with different transformations are used as inputs. Obtained results are evaluated per audio file, enabling the obtention of particular conclusions about different animal species. While the detection for some species presents encouraging results using only these first 5-shots as training data, all the tested algorithms are unable to successfully learn how to properly detect the blackbird sounds of the dataset.

Index Terms— Bioacoustics, Few-shot learning, Prototypical networks, Acoustic Event Detection, Sound Event Detection

1. INTRODUCTION

Supervised machine learning methods aim at categorizing data from a training set containing (extensive amounts of) labeled data [1]. The performance of these techniques is typically evaluated with a test dataset that incorporates data samples that not only belong to the same categories as the training set but also adhere to a similar statistical distribution [2]. Since the early stages of artificial intelligence in the 1950s, such approaches have demonstrated promising results across diverse fields, including healthcare, computer vision, robotics, and finance, among many others. However, pursuing better accuracy and performance results, building more robust systems and processing an ever-increasing amount of features, has driven modern approaches to supervised machine learning (i.e., deep learning [3]) to be astonishingly data hungry [4, 5]. This data hungriness is especially concerning in those applications in which obtaining a high volume of labeled data to build a training dataset is unfeasible and/or the computational resources for processing all the training data are unavailable [6]. Recently, this situation has motivated the conception of what has been coined as few-shot learning paradigm: an alternative approach to current data-hungry supervised learning techniques that aims at building reliable systems with a dramatically low number of labeled training examples [6].

Few-shot learning can be viewed as an effort to emulate the innate ability of humans to leverage previously acquired knowledge when learning new concepts [7, 6]. For instance, learning to ride a motorbike may require less training if an individual already knows how to ride a bicycle. Traditional methods for few-shot learning aim to take advantage of prior knowledge about certain categories (e.g., bicycle riding in the previous example) in order to learn new ones (e.g., motorbike riding in the previous example) [6]. Interestingly, this machine learning approach has attracted a lot of interest in the field of bioacoustics, particularly for tasks related to sound event detection or species classification [8]. In this domain it is very common to encounter large acoustic datasets that are very time consuming to annotate and contain highly imbalanced classes (i.e., events with with infrequent occurrences versus highly recurrent events) [8].

Typical approaches to few-shot learning consist of using pre-trained systems with a (large) set of known classes and re-training them with few—usually between two and five—shots (i.e., examples) by means of different algorithms such as meta-learning and/or prototypical networks [8, 9]. These algorithms are still data hungry [5] and strongly rely on the particular tricks and data used in this pre-training process [8]. The purpose of this work is to explore the benefits of skipping the pre-training stage in few-shot learning for acoustic data and solely training the system with five shots of data (as positive samples) plus the silence surrounding each of them (as negative samples). To obtain reference values, this work has been contextualized in the Task 5 [10] of the DCASE Challenge 2023 - Few-shot bioacoustic event detection[1] that challenges participants to detect and classify vocalizations of animals using five examples (i.e., shots) of each one of the species. For the sake of this work, the Training set provided by the challenge organizers has been left aside on purpose and different classifiers (i.e., ResNet, ProtoNet, and XGBoost) have been trained using the aforementioned five shots from the Validation set. More specifically, every audio file has been used to train a model. Obtained results have been compared to the DCASE baseline for Task 5 that features a prototypical network. This work enables researchers to assess and quantify the benefits—in terms of F1-score—of the pre-training process in few-shot learning for this particular challenge.

The remainder of this paper is organized as follows. Section 2 describes the methodology for data collection and the selection of the classifiers. Next, Section 3 presents the experimental results and their comparison with the DCASE baseline. Finally, Section 4 concludes the paper.

[1] https://dcase.community/challenge2023/task-few-shot-bioacoustic-event-detection

2. METHODOLOGY

This section delves into the methodology employed in this few-shot learning study. We start by exposing the data collection as well as the preprocessing steps. Moreover, we introduce the experimental setup and the learning methods implemented. To end up, we show the prototypical loss used in the experimentation and its mathematical sense in order to classify every event.

2.1. Data Collection and Preprocessing

Data used in this study are obtained from the Validation set of Task 5 DCASE Challenge 2023 - Few-shot bioacoustic event detection. The dataset specifically focuses on animal species detection using only five positive examples. The audio files are collected from various sources and are annotated with the corresponding species labels. Available data are split into three datasets: Training, Validation, and Test. Note that classes in the Validation set are not available in the Training set. In the Validation set, unlike in the Training Set, only positive or negative labels are considered. That is, there is no more than one species per audio file. Therefore, the objective will be to train a model that is able to discern whether a given event is a vocalization or not. Note that in this exploratory work, the Training set is intentionally neglected, and only the Validation set is used for training the learning methods. This leads us to an "extreme" few-shot learning where one model is created and trained for each audio file with the task of detecting the corresponding vocalization. Also, it is worth mentioning that the Test set has not been used as the complete annotations are not publicly available.

Before conducting the experiments, we have conducted some preprocessing steps. This involves computing the first five positive event spectrograms labeled with positive class, as well as five negative samples. Negative spectrograms are computed from intervals of silence or noise between the first five positive vocalizations of a given duration. All spectrograms are equally sized and computed using the duration of the smaller known positive or negative sample in the few-shot samples of each audio. Figure 1 illustrates an example of this preprocessing step. In Figure 1, the smallest sample is the 4th negative. As we are using the minimum duration event as window size for obtaining the spectrograms, larger events will result split in more spectrograms, so the model may be trained with more than 5 positive and negative spectrograms belonging to the same sample. To avoid class imbalance, the number of positive and negative spectrograms is always the same, being the class that presents less samples the one that limits the amount of data of each category.

2.2. Learning Methods

Three different learning methods have been employed in this study to explore the benefits of using only the initial five shots of audio data in the Validation set for training:

2.2.1. *ResNet Architecture with Prototypical Loss*

The ResNet architecture [11], a popular deep neural network, is utilized in combination with the prototypical loss function. This approach aims to learn a feature representation space where examples from the same category are grouped together. The ResNet model is initially pre-trained on a large-scale dataset (ImageNet) and then fine-tuned using the limited training data from the first five shots (positive and negative) of the audio files.

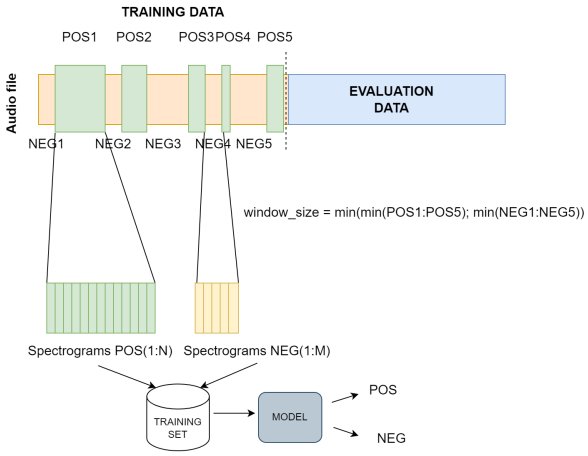

Figure 1: Extraction of positive (POS) and negative (NEG) samples from an audio file.

2.2.2. *ProtoNet*

ProtoNet [12] is a few-shot learning algorithm that builds prototype representations for each category based on a few labeled examples. It learns to classify new instances by computing similarity measures between the prototypes and the query samples. The ProtoNet uses an encoder, which is composed of multiple convolutional blocks. Each convolutional block includes a convolutional layer, batch normalization layer, ReLU activation function, and a max pooling layer. These layers are applied sequentially to the input data, transforming it and extracting meaningful features. The number of convolutional blocks can vary, but in this architecture, there are four convolutional blocks. In this study, a ProtoNet is trained with the initial five shots for each audio file and the first five computed negative samples.

2.2.3. *XGBoost Classifier*

XGBoost [13] is a gradient-boosting framework that is known for its high performance in various machine learning tasks. In this work, we aim to train a XGBoost classifier to learn from the first five shot spectrogram patterns as well as from the first five silences and make predictions on new instances.

2.3. Prototypical Loss

The prototypical loss, which has been used for training the ResNet and the ProtoNet, is a mathematical formulation used in few-shot learning tasks. Its objective is to train a model that can effectively classify new instances from unseen classes with only a small number of labeled examples. In this loss function, support examples are selected for each class in the Validation set. These support examples are used to define the characteristics of each class. The support examples of each class are averaged together to create a prototype representation, which serves as the centroid or central point of the support examples for that class (see Equation 1).

$$\mathbf{c}_j = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}_{ij} \tag{1}$$

Audio File	Precision (%)	Recall (%)	F1-score (%)
Overall ResNet	12.19	53.95	18.13
Overall ProtoNet	33.41	66.15	37.30
Overall XGBoost	31.19	64.26	36.53
Overall DCASE Baseline	22.1	49.01	28.31

Table 1: Overall percentage of Precision, Recall and F1-score of the 3 evaluated models and the DCASE Prototypical network baseline.

The remaining examples for each class, which were not used as support examples, are considered as query examples. The goal is to classify these query examples based on their similarity to the prototypes. In this case, for the similarity, Euclidean Distance is used. The prototypical loss is obtained by computing the mean log probability of the negative distances mentioned early for each ground truth class (see Equation 2). This loss encourages the model to assign high probabilities to the correct classes for the query examples. That is, this loss helps the model to project samples in an embedding space where query samples should lay near its ground truth prototype. In addition to the classification loss, a regularization term is added to the loss function. This term promotes compactness in the prototype representations by penalizing their norm.

In Equation 2 N_q represents the number of query samples. The numerator represents the exponential of the distance between the model output for query sample i and its corresponding prototype c_k. The denominator is formed by the sum of the exponential of all minus distances between query sample i and the rest of prototpyes. Finally λ is the regularization term that multiplies the norm of prototypes set.

$$\mathcal{L} = -\frac{1}{N_q}\sum_{i=1}^{N_q}\log\left(\frac{\exp(-\mathbf{d}(f_\phi(x_i), c_k)}{\sum_{k'}\exp(-\mathbf{d}(f_\phi(x_i), c_{k'})}\right) + \lambda\|\mathbf{c}\| \quad (2)$$

3. EXPERIMENTAL RESULTS

This section explains which metrics have been used and the obtained results of the experimental evaluation.

3.1. Performance Evaluation Metrics

To assess the performance of the learning methods, the following evaluation metrics are employed: precision, recall, and F1-score. Precision and recall assess the algorithm's ability to correctly classify positive instances and retrieve all relevant instances, respectively. The F1-score combines both precision and recall into a single metric. For computing those metrics, the True Positive, False Positive and False Negative rates of each audio file were obtained. The individual metrics of each audio file of the dataset have been calculated using the code provided for Task 5 2023 of the DCASE challenge, which is explained in [10]. After obtaining the individual metrics for every audio file, the metrics were averaged to obtain an overall score and thus be able to compare the different models.

3.2. Results and Analysis

Table 1 provides an overview of the performance of our three models. ResNet achieved a precision of 12.19%, indicating a poor ability to correctly identify positive instances. The recall score of 53.95% suggests that it captured a moderate proportion of relevant instances. The resulting F-measure was 18.13%, reflecting its overall performance.

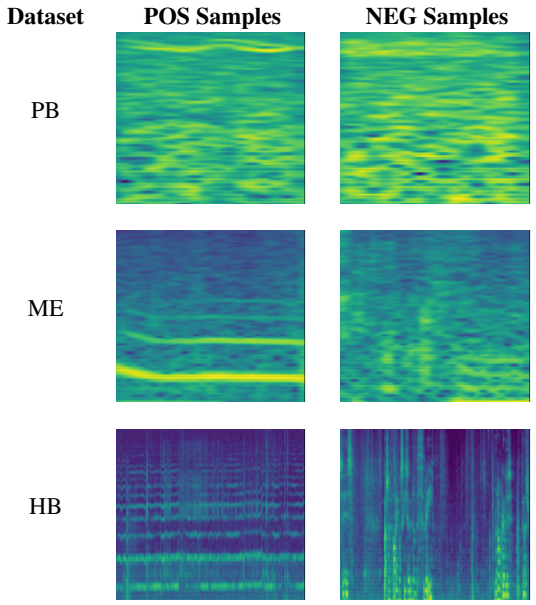

Figure 2: Positive (POS) and negative (NEG) spectrogram samples for each one of the datasets (PB, ME, HB) of the Validation set.

ProtoNet performed better than ResNet, with a precision of 33.41% and a recall of 66.15%. This means ProtoNet had a higher ability to identify positive instances and capture relevant instances. As a result, it achieved an F-measure of 37.36%.

XGBoost showed similar performance to ProtoNet, with a precision of 31.19% and a recall of 64.26%. Its F-measure was 36.53%, indicating a very similar effectiveness to ProtoNet. However it is important to point out that XGBoost requires less than the half of the training time than the ProtoNet and, also, it can be trained on a CPU with a reasonable amount of time.

Furthermore, it is worth noting that all the models presented in this study achieved higher recall metrics compared to the Prototypical network Baseline provided by DCASE Task 5 (49.01%). However, in terms of precision and F-measure, only the ProtoNet and XGBoost models outperformed the DCASE baseline that was pre-trained with the Training dataset.

In summary, ProtoNet and XGBoost outperformed ResNet in terms of precision, recall, and F-measure, with ProtoNet achieving the highest F-measure among the three models. Generally, these models are thought to be deployed in low-complexity edge devices that can be trained and used in a specific environment. For that purpose, ProtoNet is easier to deploy in a low-complexity device due to its simpler architecture and lower computational requirements. It has fewer parameters and can run efficiently on devices with limited resources. On the other hand, at inference time, XGBoost requires more computational resources and may be more challenging to deploy on low-complexity devices.

In terms of a per-audio analysis, the F1-scores for each model and audio file have been computed and summarized in Table 2. All three presented models struggle at detecting correctly the PB dataset (blackbirds). By far, this dataset presents the worst results out of the three datasets, As it can be observed, the highest score is obtained

Dataset	Animal	Audio File	F1-score ResNet(%)	F1-score ProtoNet (%)	F1-score XGBoost (%)	F1-score Baseline (%)
PB	Blackbirds	BUK1_20181011_001004.wav	0.53	1.26	0.41	2.09
		BUK1_20181013_023504.wav	0.11	0.22	0.14	5.72
		BUK4_20161011_000804.wav	0.17	0.37	0.14	0.35
		BUK4_20171022_004304a.wav	4.04	0.49	0.46	19.35
		BUK5_20161101_002104a.wav	7.68	1.88	1.88	7.67
	Song Thrush	BUK5_20180921_015906a.wav	0.14	0.21	0.21	3.38
ME	Meerkats	ME1.wav	13.45	4.32	2.04	3.48
		ME2.wav	56.25	29.14	46.38	19.51
HB	Mosquitos	R4_cleaned recording_13-10-17.wav	39.08	78.98	70.27	32.43
		R4_cleaned recording_16-10-17.wav	17.92	60.24	57.83	58.33
		R4_cleaned recording_17-10-17.wav	15.70	67.40	64.81	10.37
		R4_cleaned recording_TEL_19-10-17.wav	10.86	80.00	38.03	67.54
		R4_cleaned recording_TEL_20-10-17.wav	36.20	88.47	71.79	18.18
		R4_cleaned recording_TEL_23-10-17.wav	16.03	91.93	76.24	72.48
		R4_cleaned recording_TEL_24-10-17.wav	55.88	81.15	80.91	72.32
		R4_cleaned recording_TEL_25-10-17.wav	31.09	35.61	86.27	37.65
		file_423_487.wav	4.53	35.45	46.01	59.88
		file_97_113.wav	16.67	15.35	12.84	18.93
		Overall Scores	**18.13**	**37.30**	**36.53**	**28.31**

Table 2: Percentage (%) of F1-score per audio file of the Validation set.

by ResNet with a 7.68% of F-measure. This also happens when using the DCASE Baseline, even though in that case there is an audio file that achieved an F1-score of up to 19.35%. To motivate this behaviour, Figure 2 shows an example of positive (POS) and negative (NEG) spectrograms of this dataset. As it can be observed, the PB dataset is the one that presents more noise, with the bird vocalization being almost masked by the background noise. Visually, it is even hard to distinguish the difference between the two of them (it is the yellowest flat line on the top part of the spectrogram). With the obtained spectrograms, the presence of noise in the PB audio files might have affected the models' ability to extract relevant features and make accurate predictions, resulting in the obtained lower F1-scores. Covnersely, in the ME (Meerkats) category, the ResNet model obtained an F1-score of 13.45% in one audio file, which outperformed ProtoNet (F1-score of 4.32%) and XGBoost (F1-score of 2.04%) in the same file. On the other hand, the XGBoost model performed exceptionally well in the other file (ME2.wav) with an F1-score of 29.14%, surpassing the scores of ResNet (F1-score of 56.25%) and ProtoNet (F1-score of 46.38%). In average, the three presented models obtain better results than the DCASE baseline (except for the first audio file and the XGBoost model). Finally, for the HB (Mosquitos) category, the ResNet model achieved an F1-score of 39.08%, followed by ProtoNet with 78.98%, and XGBoost with 70.27%. The F1-scores in this category indicate that ProtoNet performed better than the other two models and the baseline.

When interpreting the results, it is crucial to consider the challenging nature of the PB audio files (very short vocalizations, background noise) and the impact they had on the models' performance. In noisy scenarios, it may be necessary to explore additional preprocessing techniques or consider using specialized models or algorithms specifically designed to handle such conditions. In this work, PCEN [14] was evaluated as a possible technique to mitigate noise, but it was discarded as it did not significantly improve the results. It is also important to consider that every audio has an independent model, so this approach is highly affected by the first initial five shots for building a solid basis to predict the rest of the audio.

4. CONCLUSION

In this study, we explored the task of bioacoustic events detection using few-shot learning techniques. Every model was trained using solely the first five positive examples of animal vocalizations as well as the first five silences (where silence means absence of the species to be detected) of every audio of the Validation set of Task 5 DCASE Challenge 2023, meaning that the Training set was not used.

Three learning methods have been evaluated: ResNet, ProtoNet, and XGBoost and compared to the DCASE baseline.

The results demonstrated that ProtoNet and XGBoost outperformed ResNet in terms of precision, recall, and F1-score. ProtoNet achieved the highest F-measure among the three models, indicating its effectiveness in discerning positive instances and capturing relevant examples. This leads us to think that simpler models in terms of parameters perform better than complex ones in few-shot learning scenarios where the training examples are limited. In general, obtained results surpass the DCASE baseline.

However, it is important to note that the presence of background noise, especially in the PB dataset, supposed a challenge to the models' performance. This highlights the need for additional preprocessing techniques and specialized models to handle such challenging conditions.

Future work should focus on improving data preprocessing techniques (e.g., filtering denoising algorithms) and exploring advanced few-shot learning methods. Moreover, it should be analysed whether expanding the dataset through data augmentation results in better performance.

5. ACKNOWLEDGMENTS

The authors would like to thank the Departament de Recerca i Universitats (Generalitat de Catalunya) under Grants Ref. 2021-SGR-01396 for Ester Vidaña-Vila and 2021-SGR-01398 for Joan Navarro for the funding of HER and SmartSociety Research Groups.

6. REFERENCES

[1] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[2] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," *The elements of statistical learning: Data mining, inference, and prediction*, pp. 9–41, 2009.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] A. Adadi, "A survey on data-efficient algorithms in big data era," *Journal of Big Data*, vol. 8, no. 1, p. 24, 2021.

[5] D. Stowell, "Computational bioacoustics with deep learning: a review and roadmap," *PeerJ*, vol. 10, p. e13152, 2022.

[6] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," *arXiv preprint arXiv:2203.04291*, 2022.

[7] L. Fei-Fei, "Knowledge transfer in learning to recognize visual objects classes," in *Proceedings of the International Conference on Development and Learning (ICDL)*, vol. 11, 2006.

[8] I. Nolasco, S. Singh, E. Vidała-Vila, E. Grout, J. Morford, M. Emmerson, F. H. Jensen, I. Kiskin, H. Whitehead, A. Strandburg-Peshkin, L. Gill, H. Pamuła, V. Lostanlen, V. Morfi, and D. Stowell, "Few-shot bioacoustic event detection at the dcase 2022 challenge," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[9] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[10] 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.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[12] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[14] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2018.