Pretraining Representations for Bioacoustic Few-Shot Detection using Supervised Contrastive Learning

Ilyass Moummad, Romain Serizel, Nicolas Farrugia

PRETRAINING REPRESENTATIONS FOR BIOACOUSTIC FEW-SHOT DETECTION USING SUPERVISED CONTRASTIVE LEARNING

Ilyass Moummad[1]*, Romain Serizel*[2]*, Nicolas Farrugia*[1]*,*

[1] IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest, France,
{ilyass.moummad, nicolas.farrugia}@imt-atlantique.fr
[2] University of Lorraine, CNRS, Inria, Loria, 5400, Nancy, France,
{romain.serizel}@loria.fr

ABSTRACT

Deep learning has been widely used recently for sound event detection and classification. Its success is linked to the availability of sufficiently large datasets, possibly with corresponding annotations when supervised learning is considered. In bioacoustic applications, most tasks come with few labelled training data, because annotating long recordings is time consuming and costly. Therefore supervised learning is not the best suited approach to solve bioacoustic tasks. The bioacoustic community recasted the problem of sound event detection within the framework of few-shot learning, i.e. training a system with only few labeled examples. The few-shot bioacoustic sound event detection task in the DCASE challenge focuses on detecting events in long audio recordings given only five annotated examples for each class of interest. In this paper, we show that learning a rich feature extractor from scratch can be achieved by leveraging data augmentation using a supervised contrastive learning framework. We highlight the ability of this framework to transfer well for five-shot event detection on previously unseen classes in the training data. We obtain an F-score of 63.46% on the validation set and 42.7% on the test set, ranking second in the DCASE challenge. We provide an ablation study for the critical choices of data augmentation techniques as well as for the learning strategy applied on the training set. Our code is available on Github.[1]

Index Terms— Contrastive learning, representation learning, transfer learning, few-shot learning, bioacoustic sound event detection.

1. INTRODUCTION

Sound Event Detection (SED) is the task of recognizing sound events, including determining their onsets and offsets, as well as recognizing them. SED has many applications in bioacoustics such as monitoring of biodiversity, studying animal behavior and identifying species. Automatic bioacoustic SED provides significant value in our understanding of animal populations and their interactions, as well as individuals and their behaviors. Standard SED systems leverage supervised learning as well as semi-supervised learning (DCASE Challenge Task 4) and have shown strong results in the recent years [1, 2, 3]. Numerous works focused on bird vocalization due to availability of large bird sound datasets [4, 5]. BirdNet [6] is a notable work for bird monitoring, able to identify nearly one thousand bird species. The approach involves training a model in a supervised fashion using a vast dataset comprising over one million labeled bird recordings, using extensive data-preprocessing and data augmentation techniques.

However, such a large scale data collection for training systems is not always feasible in bioacoustics. The challenge lies not only in obtaining annotations but also in acquiring the audio samples themselves (e.g. for rare species or fields that are difficult to reach). As a consequence, bioacoustics SED is considered as a collection of numerous small-data problems, each requiring specialized systems for their individual solutions. Thus, the community of bioacoustics recasted bioacoustic SED as a few-shot learning (FSL) problem [7, 8].

FSL is a machine learning problem where a model has to learn to adapt to new classes of data unseen during training with only few labeled samples. FSL is adapted for many applications where acquisition or annotation is expensive or time consuming. The annual challenge on detection and classification of acoustic scenes and events (DCASE) organized a third edition for the task of few-shot bioacoustic sound event detection. This task focuses on SED in a FSL setting for mammal and bird vocalizations. The goal is to create a system that learns from five exemplar vocalizations (shots) to detect instances of these vocalizations in test audio recording.

Prototypical networks (ProtoNets) [9] were proposed as a baseline to solve the FSL problem of detecting animal sound events in the DCASE challenge [8]. ProtoNets, a meta-learning framework, have been state-of-the-art FSL audio systems in the recent years [10, 11]. The goal of meta-learning training is to develop models that can quickly adapt to new tasks with minimal data by simulating the test scenario within the training process. In Computer Vision, simple transfer learning methods have been shown to outperform meta-learning methods in FSL [12, 13] in several datasets such as MiniImageNet and TieredImagenet, in which case the domain shift between the training data and the few shot generalization is small enough. Here, we propose to test transfer learning to solve FSL problems for the bioacoustic SED [8].

As the generalization capability of the feature extractor is crucial for efficient transfer learning, we propose to train a model using the supervised contrastive learning framework (SCL) [14]. Numerous contrastive learning methods have been proposed in the self-supervised learning (SSL) literature [15, 16, 17], but the fundamental concept of pulling together positive pairs and pushing apart negative pairs remains the same across these approaches. The positive pairs consist of similar samples, while negative pairs consist of dissimilar samples. The selection of these pairs can be achieved through various means, such as data augmentation techniques [15] and/or utilizing class labels as in done in SCL [14]. The representa-

This work was co-funded by the AI@IMT program and the company OSO-AI.

[1]: https://github.com/ilyassmoummad/dcase23_task5_scl

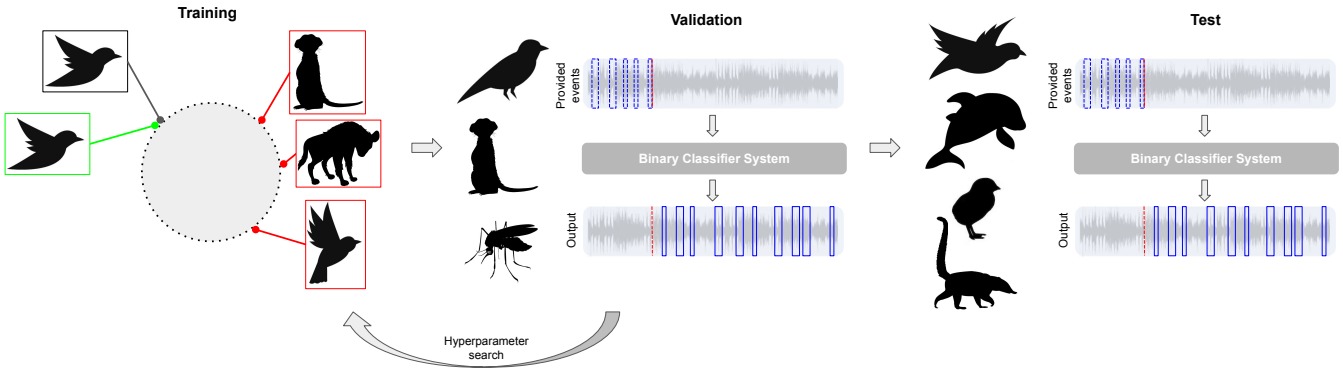

Figure 1: Overview of the proposed framework

tions learned using this framework have shown competitive transfer learning performance with SSL and cross-entropy (CE) learning on a variety of downstream tasks in vision [14]. In audio, the works of Moummad et al. [18] and Nasiri et al. [19] have demonstrated strong generalization capabilities of SCL.

Following the training of the feature extractor using SCL on the training set, the learned model is transferred to the validation set to conduct hyperparameter search. The optimal hyperparameter setting determined from this process is then employed on the test set for evaluation. In summary, our contribution revolves around the proposition of employing supervised contrastive learning to train a feature extractor that can be transferred to new few-shot bioacoustic sound event detection tasks.

2. METHOD

This section provides a comprehensive overview of the methodology employed in this study (Figure 1). Firstly, we present the SCL framework utilized for pre-training a good feature extractor model. Secondly, we describe the data augmentation techniques employed to enhance the diversity and robustness of the learned features. Finally, we detail our transfer learning strategy for adapting the pre-trained model to effectively tackle novel tasks.

2.1. Supervised Contrastive Learning

SCL consists in learning an embedding space in which the samples with the same class labels are close to each other, and the samples with different class labels are far from each other. Formally, a composition of an encoder f and a shallow neural network h called a projector (usually a MLP with one hidden layer) are trained to minimize the distances between representations of samples of the same class while maximizing the distances between representations of samples belonging to different class. After convergence, h is discarded, and the encoder f is used for transfer learning on downstream tasks. The supervised contrastive loss (SCL) is calculated as follows:

$$\mathcal{L}^{SCL} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{s \in S(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_s / \tau\right)} \quad (1)$$

where $i \in I = \{1...2N\}$ is the index of an augmented sample within a training batch, containing two views of each original sample. These views are constructed by applying a data augmentation function A twice to the original samples. $\boldsymbol{z}_i = h(f(A(\boldsymbol{x}_i))) \in \mathbb{R}^{D_P}$ where D_P is the projector's dimension. $P(i) = \{p \in I : y_p = y_i\}$ is the set of indices of all positives in the two-views batch distinct from i sharing similar label with i. $|P(i)|$ is its cardinality, $S(i) = \{s \in I : s \neq i\}$, the $\cdot$ symbol denotes the dot product, and $\tau \in \mathbb{R}^{+*}$ is a scalar temperature parameter that controls the penalty strength on hard negative samples.

2.2. Data Augmentation

Data augmentation is crucial for learning a good feature extractor as advocated by the SSL literature [15, 14, 16, **?**]. To this end, we adopt several augmentation modules derived from the audio representation learning domain [17, 20, 21]. The following augmentations are sequentially applied in the prescribed order and are iteratively employed twice on the same data, with the exception of spectrogram mixing, which is exclusively applied to a single view (based on our experimental findings, this configuration demonstrated superior performance). To demonstrate the significance of each augmentation technique, an ablation study is conducted in the subsequent section.

- Spectrogram mixing: we add background sounds using random samples from the same batch. The mixing follows: $\hat{x_1} = \alpha x_1 + (1 - \alpha) x_2$. where $\hat{x_1}$ is considered as a view of x_1 and x_2 is a random sample from the batch.
- Frequency shift: we approximate frequency shift by shifting the spectrogram upwards by few bands.
- Random crop: we crop a patch from the spectrogram along the time axis.
- Spectrogram resize: this augmentation is applied after the crop to restore the spectrogram to its original size.
- Power gain: we attenuate the power of the spectrogram by multiplying it with a coefficient sampled uniformly between 0.75 and 1.
- Additive white Gaussian noise: we add a small additive white Gaussian noise to the view

2.3. Transfer Learning

After training the feature extractor, we transfer the model to the validation and test tasks. Each audio file is treated independently as a

separate SED problem (as the challenge rules specify). We extract the features of the five positive annotated prototypes (shots) indicating the occurrence of the event of interest. We select intervals preceding the positive events as for the negative prototypes indicating the absence of the event. We train a binary classifier on these two prototypes using cross-entropy loss. The encoder layers can be either frozen or fine-tuned. We use a sliding window along the audio file (starting from the end of the fifth positive shot) to select queries for making predictions. The class activity is determined independently in each query window using the classifier. The onsets and offsets decision is made based on the precise moment when the label for the window transitions from a negative class to a positive class and from a positive class to a negative class, respectively.

3. EXPERIMENTS

3.1. Data

The bioacoustic few-shot sound event detection DCASE task development set consists of a training set and a validation set, for more details we refer the reader to the description of the task in 2022 [8] as these sets did not change from the previous edition.

3.1.1. Training

We train our system on the official training set. We select all the positively annotated segments within each audio file. We compute Mel spectrogram features with a FFT of size 512, a hop length of 128, a number of mels of 128 and a sampling rate of 22.05 kHz. Each positive annotated segment from the training set is chunked into patches of length 200 ms with 100 ms overlap. We apply min-max normalization on each patch.

3.1.2. Validation and test

For each audio file, we extract the first five positively annotated segments. The duration of these segments varies due to the wide range of animals and classes covered by the dataset. Following the approach proposed by Tang et al. [22], we determine the window length based on the mean duration of the events in the file. To compute Mel spectrogram features, we employ identical parameters and normalization technique as those used during the training phase. The shift size equals to half of the window length to predict the class for each query window along the remaining duration of the audio.

3.2. Model

We use a ResNet [23] consisting of three blocks, each comprising three convolutional layers. The feature maps of these convolutions have sizes of 64, 128, and 256, respectively. Following each convolutional layer, we apply batch normalization and a leaky rectified linear unit (ReLU) activation function. Max pooling operations are performed after each block. Specifically, we employ a 2x2 kernel for the first and second blocks, while for the third block, we use a 1x2 kernel. This choice is made to preserve frequency information by avoiding excessive pooling of the frequency bands, as suggested by Hertkorn [24].

To ensure consistent output dimensions despite varying input lengths, we incorporate adaptive max pooling at the end of the network. This pooling operation is configured to yield a desired output size of (8, 1), resulting in a latent vector of size 8 x 256 = 2048. A MLP projector is added, consisted of a hidden layer with a dimension of 2048 and an output layer with a dimension of 512.

3.3. Training details

3.3.1. Data augmentation

The spectrogram mixing coefficient α is sampled from a $\beta(5, 2)$ distribution. The frequency shift size is uniformly sampled between 0 and 10. The crop size (i.e. how much total duration is kept from the original audio) in the Random crop augmentation is uniformly sampled between 60% and 100%. Power gain augmentation is achieved by multiplying the mel spectrogram with a coefficient uniformly sampled between 0.75 and 1. The additive white Gaussian noise is incorporated by adding noise with a mean of zero and a variable standard deviation, which is uniformly chosen between 0 and 0.1.

3.3.2. Training and evaluation

We train our model from scratch on the training set using SCL framework with a temperature τ = 0.06 using SGD optimizer with a batch size of 128, a learning rate of 0.01 with a cosine decay schedule, momentum of 0.9, and a weight decay of 0.0001 for 50 epochs. After training, we discard the MLP projector and transfer the encoder to the validation and test sets by training a linear binary classifier on the pretrained representations. In this phase we used random resized crop along the time axis with a crop size ranging from 90% to 100% of the original size. We submitted four distincts systems to the challenge : freezing all pretrained layers (Frozen), or finetuning the last, two last and all layers (FineTune-1, FineTune-2 and Finetune-3). We optimize our systems using Adam optimizer with a learning rate of 0.01 for 20 epochs for the first system, and 40 epochs with a learning rate of 0.001 for the others. The selection of these hyperparameters is based on evaluation conducted on the validation set.

3.4. Results

The performance of our four systems on the validation set is presented in Table 1. For PB dataset, where events are short (therefore only few patches are available, because we divide longer events into multiple chunks), the first system outperforms the others, indicating that fine-tuning degrades the performance when only few positive patches are present. Conversely, for the HB dataset, where events tend to be longer, the third and fourth systems outperform the others. This indicates that finetuning a greater number of layers is advantageous when more positive patches are present. The second system demonstrates satisfactory performance across all datasets, outperforming the other systems across all datasets with a max F1 score of 63.46%. It is important to note that our results on the validation set exhibit significant variability, primarily attributed to the instability of our proposed cross-entropy adaptation strategy. We acknowledge this limitation and plan to address it in future work.

Table 2 displays the performance scores of our systems on the test sets. Remarkably, the ranking order of these systems on the test set aligns with that observed on the validation set. This consistency further validates the robustness and generalizability of our models across different datasets.

Table 1: Performance of different systems on the validation set; freezing all layers, fine-tuning one, two or all three layers.

System	Precision	Recall	F1-score	HB			ME			PB		
				Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Frozen	71.41	55.19	62.26	77.14	81.57	79.29	65.45	69.23	67.28	72.64	**36.17**	**48.29**
FineTune-1	**73.93**	**55.59**	**63.46**	**82.95**	82.32	82.63	67.69	84.61	75.21	**72.72**	33.33	45.71
FineTune-2	72.90	55.14	62.79	79.73	89.72	84.43	**74.60**	**90.38**	**81.73**	65.57	31.06	42.19
FineTune-all	67.08	51.58	58.32	81.20	**91.38**	**85.99**	58.75	**90.38**	71.21	65.00	27.65	38.80

*We highlight in bold the best scores for each metric

Table 2: F-score on the test sets of the different submissions

	F-score
Frozen	35.6% (35.3 - 36.0)
FineTune-1	**42.7% (42.2 - 43.1)**
FineTune-2	38.3% (37.9 - 38.7)
FineTune-all	34.4% (33.9 - 34.8)

*with 95% confidence interval

3.5. Ablation study

Table 3 presents our ablation study on data augmentation. Additionally, Table 4 compares pre-training methods : SCL, cross-entropy training (CE), and the self-supervised training method SimCLR [15], which has the same formula as SCL but without positive label pairs. We perform these studies on the validation set using the first system Frozen, where we freeze all layers, as it better captures the impact of the pre-training strategy. We use the same hyperparameter setting described in 3.3 for all experiments except for CE training where we use a learning rate of 0.0001 after thorough exploration. Additionally, we modify the training duration for SimCLR, extending it to 100 epochs. This adjustment is made to account for the longer training requirements typically associated with self-supervised approaches. To ensure reliable results, we trained the model five times on the training set and conducted five evaluations for each run, resulting in a total of twenty-five runs per experiment.

Table 3: Ablation of Data augmentation on the validation set

DA removed	Mean	[Min, Max]
Frozen (with all DAs)	56.47	[49.37, 62.39]
- Spectrogram mixing	56.59	[47.99, 64.65]
- Frequency shift	58.60	[49.73, 66.18]
- Time stretch	55.68	[49.20, 62.83]
- Power gain	56.02	[47.04, 63.01]
- Additive noise	**59.04**	**[52.68, 67.47]**

*Best score is highlighted in bold

The analysis presented in Table 3 indicates that certain data augmentation techniques have a negative impact on the model's performance. Surprisingly, these effects were not evident during the challenge submission due to the limited number of experiments conducted at that time. Notably, the data augmentation setting that yielded the highest score was the setting without the additive white Gaussian noise to the spectrogram. This finding suggests that this particular augmentation strategy was either enforcing an invariance that is not beneficial for the downstream task at hand, or that the task becomes hard given the small size of the training dataset.

We can observe from the results of Table 4 that SCL consistently outperforms both SimCLR and CE frameworks for transfer learning. The superior performance of SCL highlights its efficacy in capturing discriminative features. These findings emphasize the importance of incorporating SCL as a powerful framework for advancing feature representation learning, particularly for enhancing transferability in downstream tasks.

Table 4: Ablation of the pretraining methods on the validation set

Method	Mean	[Min, Max]
CE	51.96	[43.013-57.42]
SimCLR	50.89	[39.28-57.41]
SCL	**56.27**	**[49.37, 62.39]**

*Highest F-score is highlighted in bold

4. DISCUSSION AND PERSPECTIVES

In this study, we have provided a comprehensive description of a simple approach for bioacoustic few-shot sound event detection. We have detailed the methodology behind the systems we developed and submitted for the DCASE 2023 challenge task five. Our approach involves pretraining a feature extractor using supervised contrastive learning and data augmentation on the training set, followed by training binary classifiers on positive and negative prototypes for each audio file in the validation/evaluation sets. We proposed four systems. The first system, which utilized a linear classifier on frozen representations, demonstrated the robustness and transferability of the learned features. When fine-tuning the last layer (the second system) or the last two layers (the third system), the performance is increased. However, our current adaptation strategy, involving training classifiers on available shots, showed performance instability. We also note the gap in performance between the validation and the test sets. HB validation dataset is made of controlled lab recordings, which may make the detection easier, while PB recordings are in the wild with noisy background. Settings of the test set are more close to PB than HB [25]. To address the limitation and instability of our approach, future work will explore more effective adaptation techniques such as meta-learning. Notably, the winning systems in the 2022 and 2023 editions of the DCASE bioacoustic few-shot sound event detection challenge (Tang et al. [22]; Du et al. [26]) employed a frame-level approach, offering a higher time resolution capability compared to our window-level approach. Exploring the frame-level approach, as well as a proposal-based approach [27] for detecting variable length temporal regions of interest, which has not been previously investigated in this task, will be considered for future research. Combining representation learning (meta-learning, self-supervised learning, or supervised learning) is a promising direction for learning useful representation leveraging knowledge from large data, that can transfer well to new tasks.

5. REFERENCES

[1] T. K. Chan and C. S. Chin, "A comprehensive review of polyphonic sound event detection," *IEEE Access*, vol. 8, pp. 103 339–103 373, 2020.

[2] A. Dang, T. H. Vu, and J.-C. Wang, "A survey of deep learning for polyphonic sound event detection," in *2017 International Conference on Orange Technologies (ICOT)*. IEEE, 2017, pp. 75–78.

[3] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[4] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2019.

[5] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks." in *DCASE*, 2018, pp. 143–147.

[6] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000273

[7] I. Nolasco, S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, *et al.*, "Learning to detect an animal sound from five examples," *arXiv preprint arXiv:2305.13210*, 2023.

[8] 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.

[9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] C. Heggan, S. Budgett, T. Hospedales, and M. Yaghoobi, "Metaaudio: A few-shot audio classification benchmark," in *Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings, Part I*. Springer, 2022, pp. 219–230.

[11] Y. Wang, N. J. Bryan, J. Salamon, M. Cartwright, and J. P. Bello, "Who calls the shots? rethinking few-shot learning for audio," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 36–40.

[12] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 266–282.

[13] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, and V. Gripon, "Easy—ensemble augmented-shot-y-shaped learning: State-of-the-art few-shot classification with simple components," *Journal of Imaging*, vol. 8, no. 7, p. 179, 2022.

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[16] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[17] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 371–375.

[18] I. Moummad and N. Farrugia, "Learning audio features with metadata and contrastive learning," *arXiv preprint arXiv:2210.16192*.

[19] A. Nasiri and J. Hu, "Soundclr: contrastive learning of representations for improved environmental sound classification," *arXiv preprint arXiv:2103.01929*, 2021.

[20] L. Wang and A. v. d. Oord, "Multi-format contrastive learning of audio representations," *arXiv preprint arXiv:2103.06508*, 2021.

[21] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Exploring pre-trained general-purpose audio representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 137–151, 2022.

[22] J. Tang, Z. Xueyang, T. Gao, D. Liu, X. Fang, J. Pan, Q. Wang, J. Du, K. Xu, and Q. Pan, "Few-shot embedding learning and event filtering for bioacoustic event detection technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] M. Hertkorn, "Few-shot bioacoustic event detection : Don ' t waste information technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[25] I. Nolasco, S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, *et al.*, "Learning to detect an animal sound from five examples," *arXiv preprint arXiv:2305.13210*, 2023.

[26] G. Yan, R. Wang, L. Zou, J. Du, Q. Wang, T. Gao, and X. Fang, "Multi-task frame level system for few-shot bioacoustic event detection," DCASE2023 Challenge, Tech. Rep., June 2023.

[27] P. Wolters, C. Daw, B. Hutchinson, and L. Phillips, "Proposal-based few-shot sound event detection for speech and environmental sounds with perceivers," *arXiv preprint arXiv:2107.13616*, 2021.