UNIVERSITY
OF OSLO

Master's thesis

Unsupervised Meta-Embedding for Bird Songs Clustering in Soundscape Recordings

Joachim Poutaraud

Music, Communication and Technology (MCT) programme
30 ECTS study points

Department of Musicology (IMV)
Faculty of Humanities

Spring 2023

Abstract

Amazonian forests are threatened by numerous anthropogenic pressures not visible by satellite imagery, such as over-hunting or undercover forest degradation. Knowledge of the effects of these degradations is essential for an effective local conservation policy. However, these effects can only be assessed using quantitative methods for monitoring biodiversity in the field. In recent years, ecoacoustics has offered an alternative to traditional techniques with the development of Passive Acoustic Monitoring (PAM) systems allowing, among other things, to automatically monitor species that are difficult to identify by observers, such as crepuscular and nocturnal tropical birds. Although the use of such systems makes it possible to acquire large sets of data collected in the field, it is often difficult to process these data because they generally represent several thousand hours of recordings that need to be annotated and validated manually by an expert with in-depth knowledge of the phenology and behavior of the species studied. The objective of this thesis is to develop a new method to facilitate the work of ecoacousticians in managing large unlabeled acoustic datasets and to improve the identification of potential new taxa. Based on the advancement of Meta-Learning methods and unsupervised learning techniques integrated into the Deep Learning (DL) framework, the Meta Embedded Clustering (MEC) method is proposed to progressively discover and improve the inherent structure of unlabeled data.

Acknowledgements

I would like to acknowledge and give major thanks to my supervisor Stefano Fasciani. His guidance and advices helped me to set a framework and give meaning to my ideas throughout the writing of this project. I would also like to thank my co-supervisors Jérome Sueur and Sylvain Haupert, as well as the entire EcoAcoustics Research (EAR) team, for their incredible curiosity, sharing and humanity. I am also thankful to my family and Maria for their continuous support and understanding throughout the writing of this thesis. Finally, I would like to thank the Parisian blackbirds (*Turdus merula*) and wrens (*Troglodytes troglodytes*) who accompanied the last months of writing this thesis with their melodious and spring-like songs.

Contents

List of Tables

Chapter 1

Introduction

Due to its ubiquity and ability to efficiently transfer information over long distances, humans have often studied acoustic communication to understand animal interactions within ecosystems. Although animals communicate using a variety of sensory channels (i.e. optical, acoustical, chemical, tactile, and electrical), acoustic signals have the advantage of allowing rapid transfer of information over long distances, through obstacles, in the darkness, and without leaving traces.

As a multidisciplinary field of study, bioacoustics aims to evaluate the impact of acoustic signals in several biological contexts. In particular, one can study the intrinsic properties of the acoustic signal, its properties of propagation and transmission in the environment, and the production mechanisms used by animals to communicate. The emergence of new technologies and the process of numerical computation towards the end of the 20th century allowed bioacoustics to compete with manual analysis [1] with the development of new automated methods for the analysis of acoustic signals, such as Passive Acoustic Monitoring (PAM) systems [2] for the study and classification of species. Because the visibility of the species were not necessarily crucial for these studies (e.g. challenging scenarios such as night conditions, deep ocean), new methods became convenient in detection, classification, and localization of sound sources. In this way, population density and the impact of various anthropogenic pressures on population dynamics could be estimated remotely [3]. The mentioned technological innovations provided a foundation for the development of a scientific field we today known as ecoacoustics. Compared to earlier studies on animal communication, this recent scientific discipline aims to analyse the ecological rather than behavioral issues of animal communication [4].

1.1 Motivation

Ecoacousticians deals with rich and diverse sound typology analysis comprising a variety of acoustic units (e.g. impulsive, harmonic, iterative). Indeed, the sounds produced by most mammals have varying degrees of loudness and pitch, and different abilities to modulate in frequency for occupying different acoustic niches. Some birds are even capable to alternate between noisy and harmonic sounds or relatively pure tones using resonance in their vocal tract to enhance the energy of the fundamental [5]. As an example, a time-frequency representation of the particularities of the European Greenfinch's song is presented in Figure 1.1. In general, ecoacousticians do not focus

Figure 1.1: The song of the European Greenfinch. Top: A picture of the European Greenfinch (credits © Rogério Rodrigues). Bottom: Spectrogram of pure tone and complex call of the European Greenfinch.

on specific species but proceed with large time and space recordings where several species communicate simultaneously and at the same site. These so-called *soundscape recordings* can contain birds, insects, amphibians, and mammal recordings for terrestrial habitats, but also mammals, crustacean, and fish sounds for aquatic environments (biophony). Furthermore, biotic sound sources can be masked by abiotic sounds sources, that is, sound elements such as wind through the trees or water flows (geophony) or human produced sounds often coming from machines (anthropophony). Altogether, these recordings generate highly dynamic and complex sound scenes with a complex mix of sound sources. The latter leads us to one of the stated challenges in ecoacoustics, namely the decomposition these recordings and the identification of different sound sources to infer proper ecology information such as the absence/presence of target species. A primary motivation behind this thesis is therefore to propose a framework useful for a better understanding and visualization of highly dynamic and complex sound scenes. The goal is to facilitate the work of ecoacousticians in their management of acoustic data and identification of potential new taxa, by proposing tools that hopefully can facilitate the issues of discovering and gradually improving the inherent structure of unlabeled data.

1.1.1 Research Objectives

A recurring problem in ecoacoustics projects is the lack of large labeled datasets to train models for sound source identification. This is particularly problematic when (i) diversity is particularly high (e.g. inter-tropical regions), (ii) national biodiversity inventories could not be carried out yet (e.g. in developing countries) [6], or (iii) when rare species are targeted (e.g. rare nocturnal bird species) [7]. It is therefore still necessary to develop methods to facilitate the building of identification models based on the collection of specific training data at a local or regional scale.

The main research objective of this thesis is to improve ecoacoustics research by tackling the problem of missing large datasets, namely: *How can we get around the problem of lack of large datasets in challenging acoustic environments?* In our case, the problem is related to the image classification of bird songs known with few reference vocalizations for each species. This problem of learning from a small number of examples is often referred to as Few-Shot Learning (FSL). Note that we refer here to image classification since we analyze time-frequency representations of bird songs in two dimensions (e.g. spectrograms). The collection and annotation of datasets for the classification of animal vocalizations are indeed very difficult because the

presence of species in the recordings can be unevenly distributed over time, and the associated datasets are generally very large. Therefore, a complete manual annotation of the field recordings is not practical because extremely time-consuming, and it also requires an expert that can identify the sound of the target species with a repertoire that can sometimes be very complex. Challenges in developing datasets are a key hindrance preventing the development of algorithms and models to automatically identify rare bird species using computational techniques in ecoacoustics. Based on these observations, we propose to investigate the viability of the Meta-Learning framework for the Few-Shot Image Classification problem in the first step. This framework assumes that it is not always possible to construct a set with a sufficient number of samples to train a machine learning algorithm. In a second step, we propose to find a way to gradually improve the quality of data clustering for unlabeled samples to improve the visualization of the inherent structure of the data. For this purpose, we address the following research questions.

1.1.2 Research Questions

Q1: How well does episodic training improve the performance of a Meta-Learning algorithm compared to classical training? Meta-Learning algorithms are usually trained episodically (i.e. N-way-K-shot). In this thesis, the objective is to compare the performance of episodic training against classical training to determine the best-performing method for the Few-Shot Image Classification problem.

Q2: To what extent can Meta-Learning algorithms fine-tuned on pseudo-labeled data classify classes that were not used during training? Meta-Learning algorithms have a good generalization capacity when trained on *labeled* data. Moreover, they are easily adaptable to the Transfer Learning (TL) task, which makes them easily adaptable to new tasks. Based on these observations, the goal is here to evaluate the ability of Meta-Learning algorithms trained on *pseudo-labeled* data to generalize on classes that have not been seen during training. Pseudo-labels refer here to data that has been automatically labeled by a clustering algorithm in an unsupervised manner.

Q3: To what extent Meta-embeddings can improve the clustering quality of unlabeled data? Recent work has shown that performing data clustering in a latent space can be highly beneficial when it comes to improving the quality of the clustering [8]. Based on this assumption, it is assumed that extracting features from models which have learned to un-

derstand strategies of how to learn without prior knowledge of the data (i.e. Meta-Learning) could further contribute to improving the quality of the clustering. For this purpose, the Meta Embedded Clustering (MEC) method is proposed to gradually improve the clustering quality by allowing the model to learn meaningful representation features autonomously.

1.1.3 Research Limitations

The framework proposed in this thesis is based on techniques related to machine learning and management of acoustic data. As such, this thesis has potential limitations related to the insufficient number of data on the one hand and the ethical issue of data bias and discrimination on the other. Given the very limited access to soundscape recordings of nocturnal and crepuscular tropical bird species, it was necessary to find effective ways to circumvent this limitation. To do so, the use of the Meta-Learning framework was favored for its ability to easily adapt to new tasks. Data collection was on the other hand carried out on a collaborative database (Xeno-Canto[1]) mainly for its ease of use and access. However, this database does not allow for systematic verification of the correspondence of recordings to associated species, nor does it allow for a global representation of the extreme variety of nocturnal and crepuscular bird songs living in tropical environments. Indeed, some recordings may have been associated with bird species different from the original song because access to the database is free and collaborative and data checking is not systematic. On the other hand, given the small number of soundscape recordings available for the target species, it is common to find the same recordist name for sometimes all the recordings associated with a rare species. This can have important consequences for database creation, as the use of one recording material versus another, or the experience of the recordist in field recording, can greatly improve or degrade the performance of a machine learning model.

1.2 Contributions

The main contributions of this work are (i) the development of a completely open-source framework to perform Few-Shot Image Classification tasks with Meta-Learning algorithms fine-tuned on a pseudo-labeled dataset, and (ii) the development of a useful method to improve the clustering quality of unlabeled data to facilitate the work of ecoacousticians for the management

[1] https://xeno-canto.org/

of acoustic data and the identification of potential new taxa. To accomplish this, extensive exploration and experimentation of Meta-Learning algorithms are first performed. This includes three commonly used metric-learning based algorithms, namely: the Matching Network [9], the Prototypical Network [10], and the Relation Network [11]. Second, a comparison of the performances of classical versus episodic training is done to define the best-training method in various Few-Shot Image Classification tasks. Finally, the best-performing model and training method are used to extract meaningful latent space representations (i.e. embeddings) to improve the clustering of unlabeled data, which is further improved by fine-tuning the model on pseudo-labels generated by a clustering algorithm.

A unique dataset composed of acoustic units of nocturnal and crepuscular tropical bird species collected and segmented from the Xeno-Canto online database is built. Nocturnal and crepuscular bird species are so far underrepresented in the literature, especially in tropical environments. Therefore, contributing to the study of such species can be beneficial for better understanding their behaviors. As a result, the proposed dataset is released as an open-source and code-based dataset that can be easily downloaded with the following link: `https://github.com/joachimpoutaraud/darksound`. Finally, the objective is to contribute to the improvement of the identification and visualization of rare bird species living in tropical environments, as well as the discovery of potential new taxa based on acoustic properties only.

1.3 Thesis Structure

The framework proposed in this thesis is based on the use of computational methods developed in the fields of ecoacoustics as well as on concepts and techniques developed in machine learning. An overview of the historical evolution of the literature is presented in chapter 2. Moreover, a critical perspective on the state of the art of current research is defined in chapter 3, notably the introduction of Meta-Learning methods (section 3.1) and their use in the framework of unsupervised learning (section 3.2). Chapter 4 presents the material and methods of the proposed framework with details about the creation of the Darksound dataset (section 4.1) and the methods and evaluation criteria used in our experimental design. The methods are essentially based on the use of three pre-existing Meta-Learning algorithms (subsection 4.2.1) and on the introduction of a method aiming at iteratively improving the quality of data clustering (subsection 4.2.2). Furthermore, in chapter 5, the results of our research are presented and discussed with our initial questions

outlined in subsection 1.1.2. A critical view of the environmental considerations related to the methods used in this thesis is also developed, as well as a proposal for future work. Finally, the thesis is concluded in chapter 6.

Chapter 2

Background

This chapter provides a general overview of the computational techniques used in the field of ecoacoustics, with a particular focus on applications related to Deep Learning (DL) for classification and Deep Embedding for clustering analysis. The objective is to recontextualize the standard recipe for ecoacoustics tasks in the DL framework while introducing the different types of methods related to our research.

2.1 Computational Bioacoustics

Computational Bioacoustic Scene Analysis (CBSA), as a specific task of ecoacoustics, has recently been included in the Detection and Classification of Sound Scenes and Events (DCASE) Challenge [12], which was organized by the IEEE Technical Committee on Audio and Acoustic Signal Processing. An overview of the different tasks proposed by the DCASE Challenge over the years is presented in Table 2.1. This challenge was designed around the use of *automatic listening* systems with two types of classification tasks, involving the acoustic scenes and the sound events. On the one hand, the challenge included the recognition of the type of acoustic scene, and on the second hand, the detection and classification of sound events within the acoustic scene. CBSA gathers a vast field of applications in ecoacoustics with, for example, the monitoring and the conservation of populations through the detection, classification, and localization of individuals representing species. The common point of these tasks is related to the numerical computation process which is crucial since it allows the implementation of automated studies on a large scale and over long periods, unlike traditional methods which are relatively expensive, and limited in space and time [13, 14]. Moreover, the numerical computation process allows the development of new automated

IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events					
DCASE Tasks	**2018**	**2019**	**2020**	**2021**	**2022**
Task 1	Scenes				
Task 2	Tags		Monitoring		
Task 3	Birds	Localization			
Task 4	Large-scale	Domestic			
Task 5	Monitor	Urban		Bioacoustics	
Task 6				Caption	

Table 2.1: Overview of the tasks of the Detection and Classification of Sound Scenes and Events (DCASE) Challenge from 2018 to 2022.

methods for the resolution of problems related to the detection and classification of animal vocalizations. Thus, the CBSA framework allows to target (i) the simple presence/absence evaluation of species occurrence in an audio recording (i.e. weak labelisation) [15], or (ii) the exact time position of the animal vocalizations by determining the onsets and offsets (i.e. strong labelisation) [16].

2.1.1 Species Identification and Localization

In ecoacoustics, identifying animal vocalizations in audio recordings is considered to be a useful means for estimating the population density of one species with another, by setting up identification and localization methods adapted to the species under study. The estimation of population density integrates a notion of individual tracking within a group of animals and allows in particular the in-depth analysis of the main mechanisms contributing to the recognition of an individual within a group, namely: the distinction of the types of emissions and the production of individual signatures emitted only for identification [17]. However, the frequency of use of acoustic signals by species can change temporally and spatially, making individual identification and population density estimation very tricky. Indeed, to estimate population density, it is necessary to be able to first count the number of individuals by identifying them one by one to define their acoustic signatures and to localize them. Some studies have approached this but with proxy approximations [18]. In addition, some species are difficult to access and the lack of data on them may hinder their protection [17, 19]. In this regard, [5] mentions that species identification is generally well specified in developed countries, thanks to a large volume of data available for training a classi-

fier, but that it is sometimes unfeasible in developing countries or in places that are difficult to access, mainly due to the lack of accessible data. This highlights the need for different strategies (e.g. supervised vs. unsupervised methods) depending on the state of development of the country (i.e. large dataset in developed countries with lower biodiversity vs. small dataset in developing countries with higher biodiversity). Regarding the problem of species identification and recognition in developing countries, new methodological approaches related to the clustering and the visualization of acoustic units and their sequencing have recently been developed [20, 21].

Acoustic analysis of the environment can also allow to estimate the spatial location of species or tracking of known and unknown individuals. In most cases, this corresponds to the estimation of the direction of arrival of sound for the position and orientation of a microphone array. This generally requires triangulation based on the speed of sound and the relative arrival time of a sound at each of the microphones [22] and allows for the implementation of species behavior studies, species-specific activity modeling, species abundance estimates, or population density assessments. However, this is generally only feasible at short distances (of the order of a few meters) and depends on the configuration of the microphone array, the spacing chosen between the microphones, and their sensitivity.

2.1.2 Sound Event Detection

Before classifying or clustering the acoustic units of animal species, one must be able to detect them. For this purpose, it is necessary to develop methods adapted to the specific species of interest and control whether their acoustic signals are present in an audio recording. Note that an audio recording without acoustic signals does not necessarily mean the absence of a species, since many species can evolve in their ecosystem in silence. Thus, the task of Sound Event Detection (SED) can be approached in several different ways. According to [23], three broad types of categories can be distinguished.

1. Binary detection (presence/absence)
2. Sound event detection (temporal start/end)
3. Object detection (temporal and frequency start/end)

The main difference between these three categories is that the first aims only at confirming the presence/absence of a target species, while the second and third focus also on detecting the beginning and end of the corresponding

sound events. More precisely, the first detection task aims at describing the "occupancy" information of the species by detecting in a binary way its presence/absence in an audio recording. The second task aims at defining the temporal boundaries of the start/end regions of a sound event, and finally, the third detection task represents the temporal and frequency boundaries of an "object" (i.e. a sound event) in a graphical representation of the sound (e.g. a spectrogram). The three common approaches to the implementation of sound detection are illustrated in Figure 2.1.

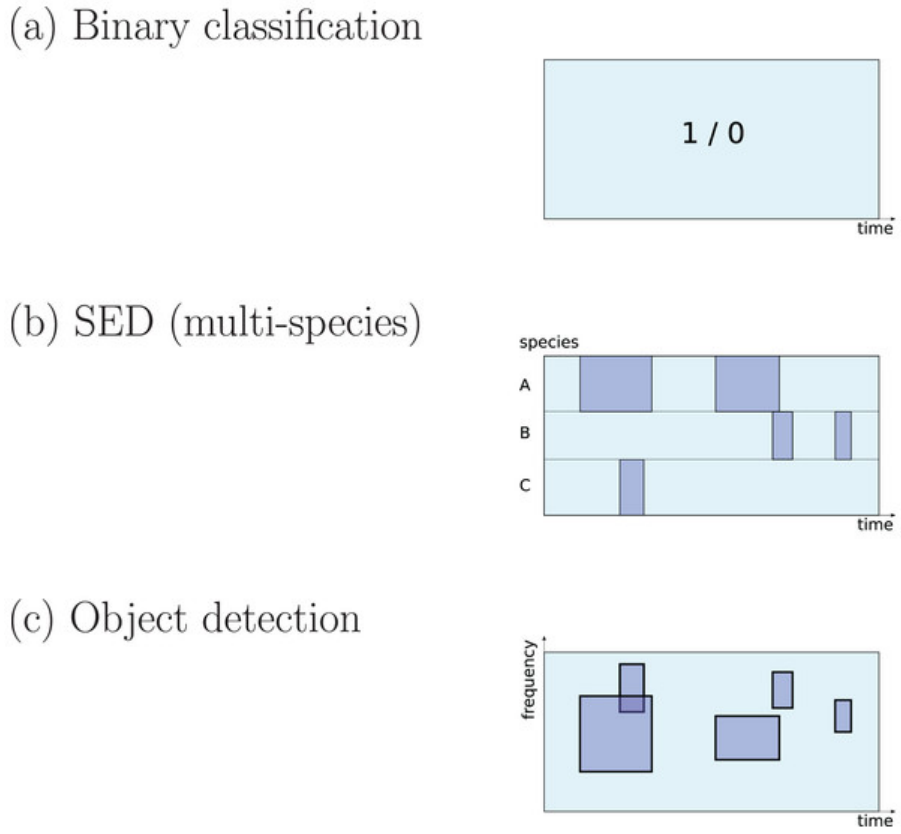


Figure 2.1: Three common approaches to implementation of sound detection. (a) Binary classification: describes the "occupancy" information of the species by detecting in a binary way its presence/absence in an audio recording, (b) SED: defines the temporal boundaries of the start/end regions of a sound event, and (c) Object detection: represents the temporal and frequency boundaries of an "object" in a graphical representation of the sound. From "*Bird detection in audio: a survey and a challenge*", by D. Stowell *et. al.* 2016b, PeerJ. (https://doi.org/10.7717/peerj-13152/fig-1)

2.1.3 Passive Acoustic Monitoring

Ecoacoustic surveys can be performed over periods ranging from a few seconds to several years. Thus, it is sometimes necessary to detect the presence of active species over very long time scales. This is particularly the case in the marine context where surveys on very large time and space scales are common, or in environments where species access is limited. In such cases, manual analysis of audio recordings appears impossible [24]. Indeed, manual analysis can be time-consuming and requires the knowledge of experts who can introduce a bias in the data annotation phase. In addition, few experts can identify animals with certainty based on their acoustic signals. Therefore, the use of automated methods appears to be an alternative solution to overcome the problems related to the detection of species over long periods. In particular, the use of Passive Acoustic Monitoring (PAM) systems such as Autonomous Recording Units (ARUs) allows the study of species 24 hours a day, every day of the year, and in several habitats simultaneously [25, 26]. Moreover, ecoacoustic surveys generally follow a two-step workflow: detection and classification (although this can sometimes be combined). This has the advantage of facilitating the learning process by applying an automatic classification only on the detected regions in the acoustic scene. The goal is here to optimize the storage and transmission of data by rejecting a large number of "negative" sound clips beforehand [23]. In general, ecoacoustics studies make extensive use of Machine Learning methods for classification. Classification is commonly applied at the species level, typically within a family of taxa, as in the BirdCLEF challenge [27]. However, it can also be applied at the scale of individuals within a particular species, for example, to estimate population numbers or to analyze interactions between individuals. Classification at the scale of individual acoustic units is relatively little considered, mainly because it is difficult to generalize to a whole species, given the large number of individual differences and differences between distinct populations.

2.2 Deep Learning for Classification

Machine learning is a field of study in Artificial Intelligence (AI) that allows a machine to "learn" from data. This field has recently revolutionized the field of ecoacoustics, in particular with the introduction of detection, classification, and clustering methods in the sub-field of Deep Learning (DL). Although DL algorithms are used for a wide range of applications (e.g. classification/regression, signal enhancement, or new data synthesis), the ma-

jority of its applications focus on DL classification methods studies. Classification is therefore considered the main method used in ecoacoustics with DL [23]. This implies different types of classification such as binary classification, multi-class classification, multi-label classification, or imbalanced classification. In this thesis, particular attention is paid to the multi-class and imbalanced classification tasks.

Multi-Class Classification

The goal of multi-class classification is to predict to which class an input example belongs with a minimum of two mutually exclusive class labels. For example, classifying the song of five different bird species is a multi-class task where the goal is to train a model able to correctly classify the vocalization of a bird based on the properties of the class associated with it. For this purpose, different architectures of DL algorithms can be considered. Most of the algorithms used for this kind of task are referred to as "*eager learners*". Meaning that they have the particularity to build a model from a training data set to make predictions on validation and test data sets. On the other hand, "*lazy learners*" are used to memorize the training data, and look for the nearest neighbor from the training data set to make a prediction. This has the disadvantage of making them very slow during prediction.

Imbalanced Classification

A common case in multi-class classification tasks is that the number of examples is unevenly distributed in each class. This problem is related to imbalanced classification, meaning that there may be more samples from one class than others in the training data. There are several approaches to tackling the imbalanced database problem. In this thesis, the most common approach is used, that is to oversample the data (although it is sometimes necessary to *undersample* the data), so that each class in the database has a similar number of samples. Popular sampling techniques that can be used for multi-class classification include:

- Simple Random Sampling
- Cluster Sampling
- Systematic Sampling
- Stratified random Sampling

2.2.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are often defined as an organized set of interconnected neurons allowing the solution of complex problems such as computer vision, machine listening, or Natural Language Processing (NLP). More precisely, the objective of an ANN is to mimic the architecture of the human brain through a combination of data inputs, weights, and biases. It is composed of a collection of connected units called artificial neurons that act as conceptual derivatives of biological neurons. Each artificial neuron receives inputs from several other neurons, multiplies them by assigned weights, adds them up, and passes the sum to one or more neurons via the use of a transfer function, as illustrated in Figure 2.2. Early work in ecoacoustics made use of

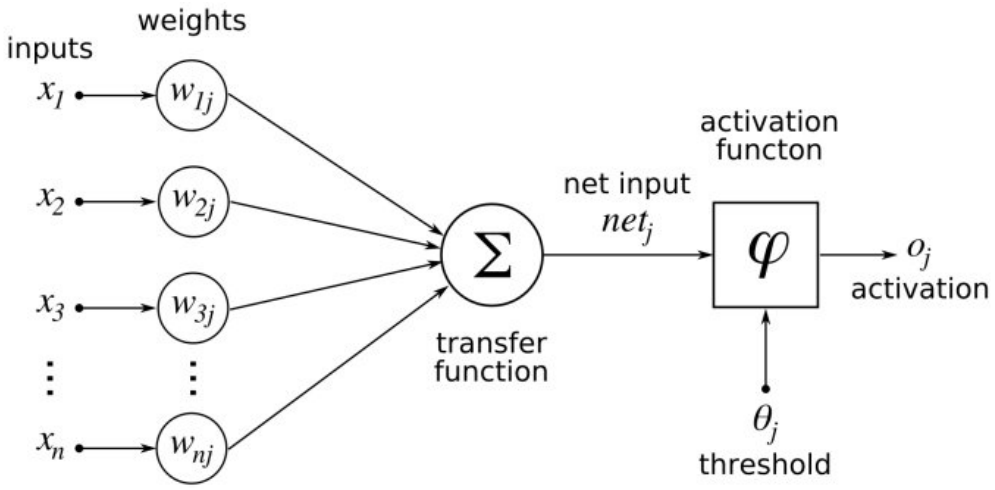


Figure 2.2: Diagram of the structure of an artificial neuron, the basic component of ANNs. "*Diagram of an artificial neuron*", 2005, av Chrislb. (https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png)

ANNs for their ability to detect animal sounds, notably with the use of basic architectures such as the Multi-Layer Perceptron (MLP) [28, 29]. The MLP is a typical example of a *feed-forward* ANN composed of a series of layers of nodes including an input layer, several hidden layers, and an output layer, as illustrated in Figure 2.3. In the context of DL, such algorithms are also known as Deep feed-forward Networks or Deep Neural Networks (DNNs) and are the quintessential DL models. The term "*feed-forward*" refers here to the fact that there are no *feedback* connections in which the model's outputs are sent back to itself. In the opposite case, we speak of Recurrent Neural Networks (RNNs) which have the particularity to include *feedback* connections.

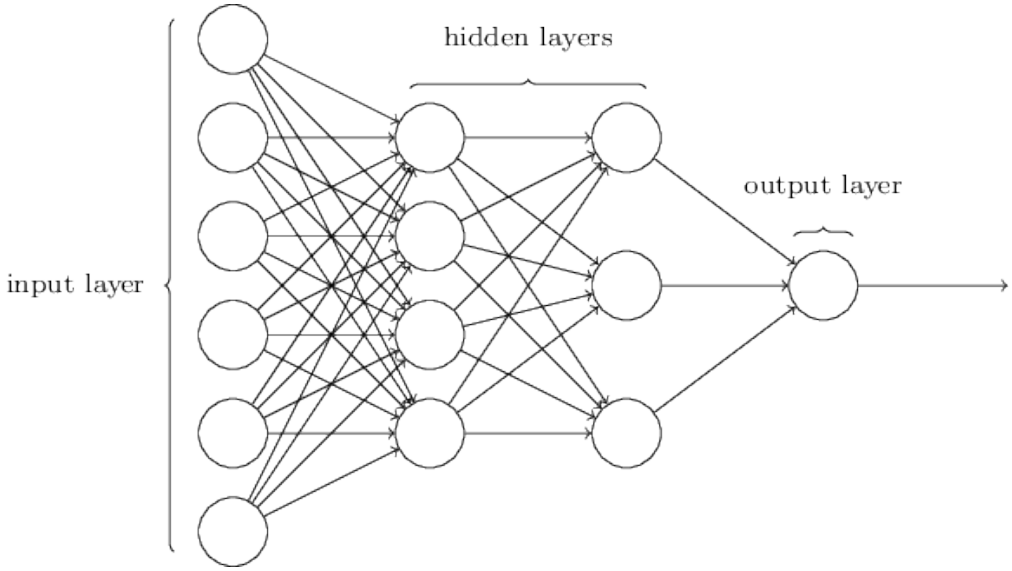


Figure 2.3: Diagram of the basic structure of a Multi-Layer Perceptron (MLP) with one input layer, several hidden layers, and one output layer. "*Multilayer Perceptron*", 2020, av David Rodriguez. (https://github.com/d-r-e/multilayer-perceptron)

Regarding the field of ecoacoustics, DNNs have proven to be very efficient for audio classification tasks and have been widely introduced in [23].

Architectures

There are many types of ANN architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Auto-Encoders (AEs), Transformer Neural Networks (TNNs) or Generative Adversarial Networks (GANs). Currently, the vast majority of audio classification tasks in ecoacoustics make use of CNNs which often outperform other ANN architectures. Nevertheless, new approaches using TNNs have recently exposed the difficulties that CNNs may have in capturing long-term relationships and global context in audio data [30]. According to [23], the earliest study using a CNN in ecoacoustics is related to the classification of 10 anuran species [31]. The use of the CNN was then widely developed, notably for the majority of works participating in the BirdCLEF challenge, which use CNN systems with spectrograms as input, including the highest scoring team. The application of CNNs in the context of ecoacoustics generally makes use of off-the-shelf CNN architectures, such as ResNet [32], VGG [33] or DenseNet [34]. These types of architectures have been made popular by their accessibility and ease

of use in the context of DL. On that matter, [23] has established a study of the standard off-the-shelf CNN architectures used in the literature according to their number of appearances in papers. The ranking of the top 5 off-the-shelf CNN architectures is presented in Table 2.2.

CNN architecture	**Number of articles**
ResNet	23
VGG or VGGish	17
DenseNet	7
AlexNet	5
Inception	4

Table 2.2: Ranking of the top 5 off-the-shelf CNN architectures according to their number of appearance in articles in 2022. From "*Computational bioacoustics with deep learning: a review and roadmap*", by Stowell *et. al.* 2022, PeerJ.

Activation Functions

When used for supervised classification, the objective of an ANN is to estimate the probability of an input belonging to a class. For this purpose, various types of activation functions can be applied to the output from a node or nodes in a layer of the network. In this thesis, the focus is mainly on the output layer of the network. Activation functions correspond to the weighted sum of all the inputs that has been weighted by the weights of the connections from the inputs to the neuron. The weighted sum of the inputs is computed as $x_1w_1 + x_2w_2 + ... + x_nw_n$, where $x_1, x_2, ..., x_n$ correspond to the inputs and $w_1, w_2, ..., w_n$ to the weights. A bias β is then added to the weighted sum $x_1w_1 + x_2w_2 + ... + x_nw_n + \beta$ to feed the computed value to the activation function φ that produces the output.

$$\varphi(x_1w_1 + x_2w_2 + ... + x_nw_n + \beta) \tag{2.1}$$

Regarding multi-class classification, the Softmax function is commonly applied on the output layer of the network to generate a normalized probability score, whose total sum of probabilities is equal to 100%, i.e. 1. Softmax function is defined in the equation 2.2.

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, 2, \ldots, K \tag{2.2}$$

It is important to bound the total sum of probabilities within a small range in order to avoid having huge weights or numbers while progressing backward or forward within the network. In addition, the use of logarithmic probabilities with the probabilities can be useful to improve the numerical performance and optimize the gradient. For this purpose, it is sometimes advisable to use the logarithm of the Softmax activation function which allows to strongly penalize the model when it fails to predict a correct class. The choice of the activation function is to be tested according to the statement of the problem that the model tries to solve.

Back Propagation

Back propagation is the method used to train ANNs, optimizing the parameters of transformations, in particular weights and biases, from the last layer to the first. This allows the ANN to learn from its errors and to correct internal parameters according to the relative importance of the contribution of each element. To correct internal parameters, a learning rate parameter is used to define the size of the corrective steps that the model takes. Then, weights that contribute the most to an error are modified more significantly than the weights that cause a marginal error. Back propagation calculates the gradient of a cost function, so that weights can be updated using gradient descent methods (e.g. Stochastic Gradient Descent (SGD)). The objective here is to converge iteratively to an optimal configuration of the weights, which represents a minima of the loss-function (ideally the global minima, although training algorithms may be stuck in sub-optimal local minima). Regarding multi-class classification, the Cross-Entropy loss, or log loss function is commonly used to measure the performance of a model that produces an output with a probability value between 0 and 1. It calculates a separate loss for each class label per observation and sums the result. It is defined in equation 2.3, where M corresponds to the number of classes > 2, y represents the binary indicator (0 or 1) if the class label c is the correct classification for observation o, and p the predicted probability observation o is of class c.

$$\mathcal{L}_{CE} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{2.3}$$

2.2.2 Acoustic Features

In the context of ecoacoustics, the magnitude spectrogram is generally used as the input of DL algorithms. This finding is established by [23] on a set of 162 articles surveyed. The advantage of the spectrogram is to represent the

intensity of the sound in a single time-frequency space. Several computational parameters affect the time-frequency representation of the spectrogram. For example, the length of the window used to compute the Short-Time Fourier Transform (STFT) or the form of the window function used (e.g. Hann window). Regarding the window length, a shorter window will provide a better time resolution of a particular sound event, whereas a longer window will provide a better frequency resolution. The problem related to the length of the window refers to a well-known dilemma from quantum physics which was stated as Heisenberg's uncertainty principle. A comparison of the impact of the window length on two Mel spectrograms is illustrated in Figure 2.4. Ac-

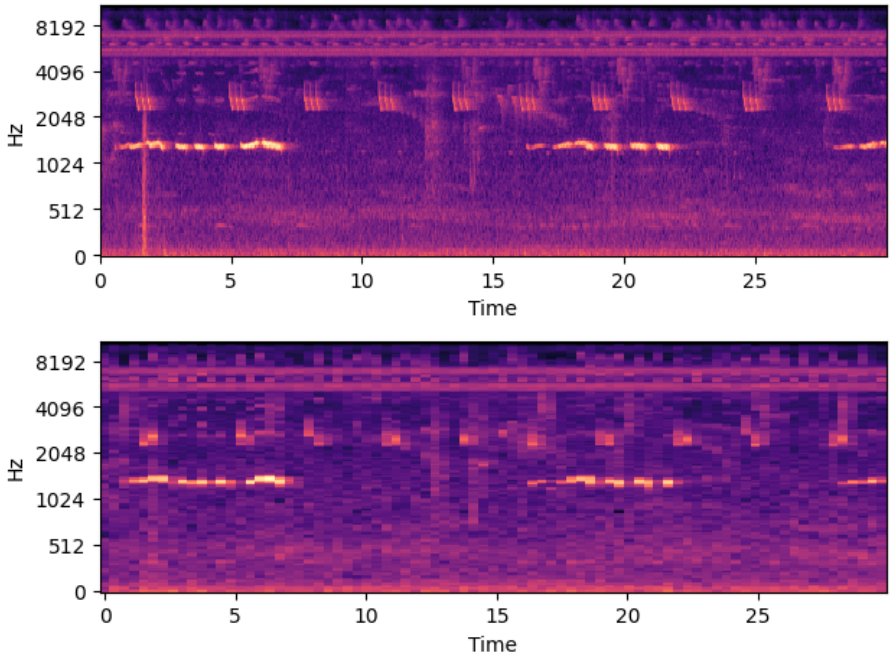


Figure 2.4: Comparison of the impact of the window length on two Mel spectrograms. Top: Mel spectrogram with a window length of 64 samples. Bottom: Mel spectrogram with a window length of 8192 samples.

cording to [35], the fine-tuning of these parameters can significantly improve the performance of DL classification models. Moreover, how the frequency axis of the spectrogram is scaled has also an important impact, although there is no strong consensus in the literature on which type of scale should be used. In the context of ecoacoustics with DL, it is usually not scaled (i.e. linear) or scaled logarithmically. When a logarithmically frequency axis is used, it is common to use either the constant-Q transform (CQT) spectrogram [36] or the Mel spectrogram [37] which are both based on the human

frequency process. The difference between the two is that the former is based on a semitone scale while the latter is based on the Mel scale. The semitone scale represents equally spaced tones on a logarithmic scale, whereas the Mel scale makes equally spaced tones sound the same distance apart, regardless of pitch. The conversion from Hertz (f) to Mels (m) is expressed in equation 2.4 and a visual difference of the CQT spectrogram compared to the Mel spectrogram is illustrated in Figure 2.5.

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \tag{2.4}$$

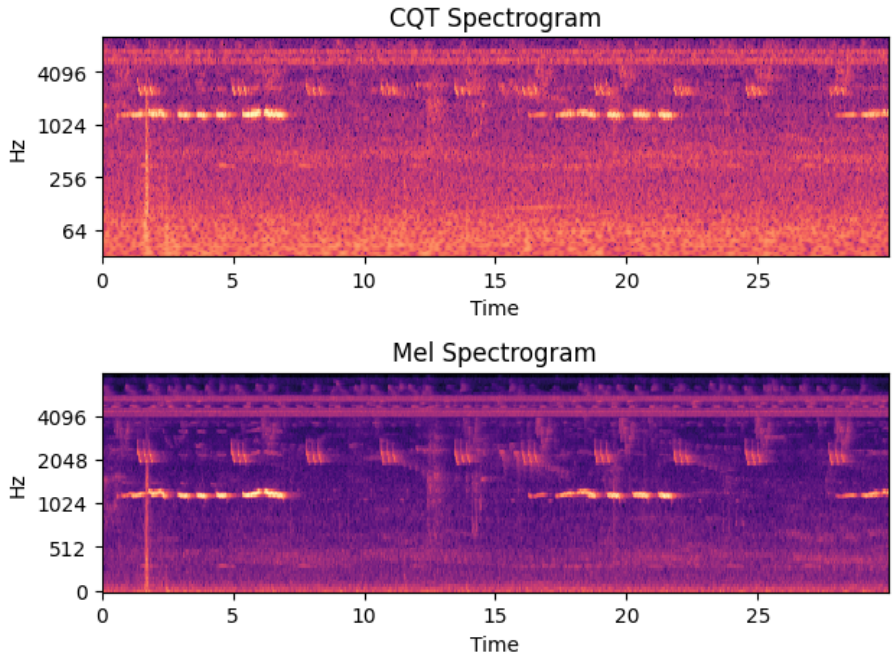


Figure 2.5: A visual difference of the CQT spectrogram compared to the Mel spectrogram. Top: CQT spectrogram. Bottom: Mel spectrogram.

That is to say, the choice of a particular time-frequency representation is not systematically appropriate to represent all aspects of the diversity of animal vocalizations. On the other hand, the image can alternatively be optimized by Per-Channel Energy Normalization (PCEN) [38], although there is no strong consensus on this. Finally, the learning process of a model considers the spectrogram in parts, dividing it into fragments of a few pixels or slices of a few seconds, on which a succession of filters creating new images (i.e. convolution maps) are applied. These successively repeated convolutions make

it possible to obtain a latent space representation (i.e. embedding), which is evaluated through several Fully Connected (FC) feed-forward layers. The classification happens at the output layer of the network, which is composed of one neuron per class, with the logarithm of the Softmax activation function estimating the probability of belonging to a class. The different steps required to classify a spectrogram using a CNN are summarized in Figure 2.6.

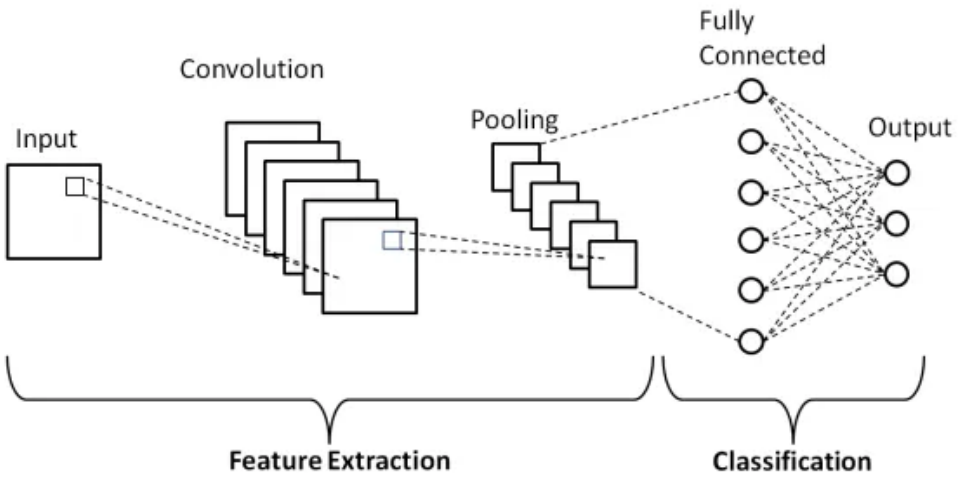


Figure 2.6: Schematic representation of the overall steps of a CNN. Features are extracted from an input image by applying a set of filters (or kernels) using a convolutional layer and reducing the dimensions of the hidden layer with a pooling layer. Output variables of the classification are obtained with a Fully Connected (FC) layer where each neuron applies a linear transformation to the input vector through a weights matrix. "*Binary Image classifier CNN using TensorFlow*", 2020, av Sai Balaji. (https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697).

2.2.3 Data Augmentation and Pretraining

Many animal vocalizations are still difficult to access today, either because they are rare species or because their vocalizations can only be recorded in environments that are difficult to access (e.g. underwater environments, tropical rain forests). Due to their rarity, it is thus difficult to have access to large datasets for the training of a DL classification model. Data augmentation and pre-training are two techniques proposed to alleviate the issue of data scarcity.

Data Augmentation

Data augmentation is a common technique to artificially increase the size of a dataset (usually the training set). In subsection 2.2, several methods related to data sampling were cited to tackle the problem of imbalanced classification. These methods also apply to problems related to data scarcity, by allowing the number of data samples to be increased by applying small modifications to create additional data samples. In the context of ecoacoustics with DL, such modifications typically include:

- Time-shifting (e.g. random backward or forward time shift)
- Soundscape synthesis (e.g. mixing "ambient sound" recorded in situ)
- Frequency equalization (i.e. slightly changing the response characteristics of the microphone by convolving it with an impulse response)
- Emulating the effects of distance by attenuating high frequencies due to air absorption
- Adding low-amplitude Gaussian noise (although it is better to add ambient sound)

Note that it is important not to adopt methods directly from image processing without considering the aspect they transform. More precisely, while it is possible to assume that neighboring pixels can belong to the same visual object in standard image processing applications, this is not true for sound images such as spectrograms. Spectral properties of sounds are non-local, therefore, the representation of the frequencies in a spectrogram is generally non-locally distributed [39]. Consequently, moving the frequencies of a bird song upwards or downwards can be artificial and extends out of natural variation. For example, applying frequency shift to bird songs could change the belonging of a species to a class. Nevertheless, if the frequency range of the repertoire of the species to be augmented is known beforehand, this can be considered.

Pretraining

Transfer Learning (TL) methods can be used as a meaningful way to reuse the backbone of models pre-trained on large datasets, preferably similar to the new task. In image classification, this is generally done to allow the fine-tuning of the last layers of the backbone to classify new categories. As a result, fine-tuning a network using TL allows one to generalize to other

categories by reusing the first layers of the network as an efficient way to solve problems related to edge detection or basic shape detection. Regarding ecoacoustics with DL, recent works using pre-trained models on the ImageNet dataset have been able to significantly improve model performance, even though the images turn out to be very different from the spectrograms [40]. Other approaches have used TL to improve the performance of audio classifiers using a one-dimensional convolutional layer. To this end, models are generally pre-trained on Google's AudioSet dataset or VGG Sound dataset [41] and allow the application of TL methods for the classification of one-dimensional audio waveforms. This type of approach can also eliminate the need to worry about the time-frequency representation problems related to Heisenberg's uncertainty principle.

2.3 Deep Embedding for Clustering Analysis

Deep Neural Networks (DNNs) can be used as feature extractors to reduce a given data space to a lower dimensional latent space. To do this, DNNs transform images into latent space representations (i.e. embeddings) that can then be used to compute a distance and to measure the similarity or the dissimilarity between the representations. Deep embeddings can also be used to learn feature representations for clustering the data on the latent space. Such an approach has notably helped improve the quality of data clustering by optimizing a clustering objective in an iterative way [8]. As a result, this makes it possible to improve the quality of the data clustering as well as the representation of the extracted features.

2.3.1 Clustering Algorithms

Data clustering techniques such as clustering algorithms are used to discover the inherent structure of the data and can improve its visualization. The advantage of clustering algorithms is that they use unsupervised learning to discover hidden patterns or clusters of data without the need for human intervention. Various methods allow to cluster elements within classes (as homogeneous as possible) but without knowing the classes a priori. In this sense, clustering algorithms aim at creating clusters with high intra-group similarity (similarity within clusters) and low inter-group similarity (dissimilarity between clusters). In the following, the four most common approaches in clustering analysis are introduced. A representative overview of the clustering algorithms is shown in Figure 2.7.

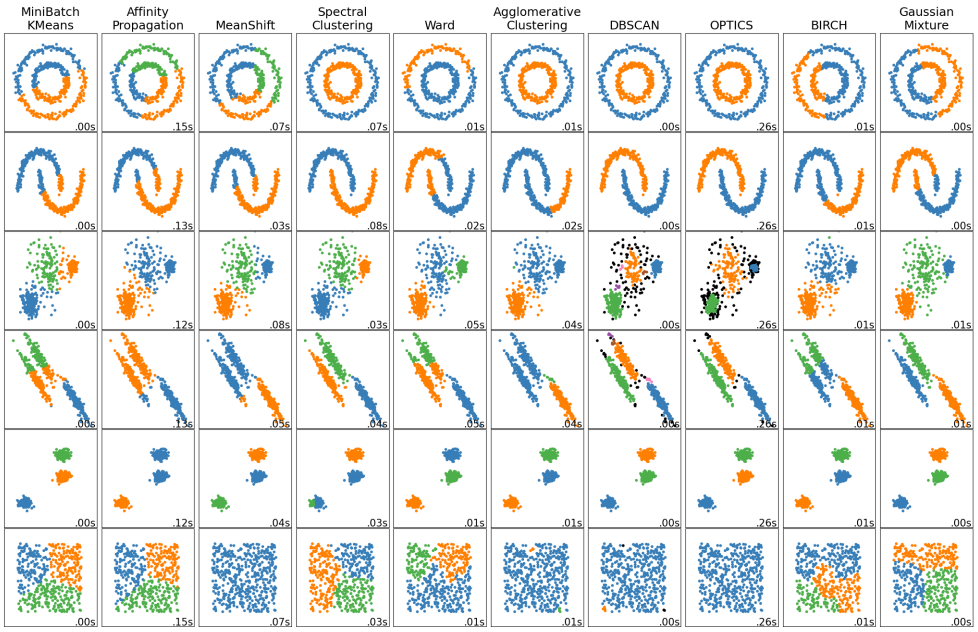


Figure 2.7: A comparison of the clustering algorithms in scikit-learn. "*Overview of clustering methods*", 2007-2023, av Scikit-Learn. (https://scikit-learn.org/stable/modules/clustering.html)

Centroid-Based Clustering

Centroid-based algorithms organize data into non-hierarchical clusters. One of the most popular centroid-based algorithm is certainly the k-means with its ability to divide observations into k clusters. However, the main drawbacks of this algorithm is that it is sensitive to noise and requires the number of clusters to be specified.

Hierarchical-Based Clustering

Algorithms based on hierarchical clustering allow the creation of a dendrogram able to illustrate how each cluster is composed. This is very well adapted to hierarchical data such as taxonomies and has the advantage of not having to specify the number of clusters beforehand.

Density-Based Clustering

Density-based algorithms are useful to merge areas of high example density into clusters. They have the advantage of not having to specify the number of clusters and they can deal with noise and keep it outside any clusters (e.g.

DBSCAN, OPTICS). However, these algorithms have difficulties with data of varying densities and high dimensions.

Distribution-Based Clustering

It is also possible to cluster data by assuming that they are composed of distributions (e.g. Gaussian distributions). Thus, if a point moves away from the center of the distribution of a cluster, its probability of belonging to the cluster is low, and vice-versa. However, this requires to know the type of distribution of the data in advance.

2.3.2 Deep Embedded Clustering

Clustering algorithms are often used in combination with dimensionality reduction algorithms that are useful to project data from a high dimensional space into a lower dimensional space. Common techniques use linear dimensionality reduction frameworks such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), or Linear Discriminant Analysis (LDA). Non-linear dimensionality reduction frameworks can also be used, notably Manifold Learning which allows to generalize linear frameworks like PCA by learning the high dimensional structure of the data from the data itself. A visual comparison of Manifold Learning methods is illustrated in Figure 2.8. Another solution to avoid the "*curse of dimensionality*" [42] is to use initialized parameters of a DNN as a feature extractor to transform the data space into a latent space. This allows the performing of initial non-linear mappings of the data that are appropriate for complex data representation. Moreover, using a lower dimensional space can be useful for clustering the data. In this regard, the Deep Embedded Clustering (DEC) method introduced in [8] has shown that passing the data through an initialized DNN to get an initial estimate of the non-linear mappings can allow optimizing a clustering objective by performing a clustering algorithm directly on the latent space, making it possible to refine the initial cluster centroids by updating the parameters of a DNN.

Clustering Loss

DEC method is defined as a clustering loss between the "*soft assignments*" (i.e. the relation between the embedded points and the cluster centroids) and a target distribution [8]. This is achieved by minimizing the Kullback-Leibler (KL) divergence iteratively. KL divergence allows to measure the difference between one probability distribution P from a second probability

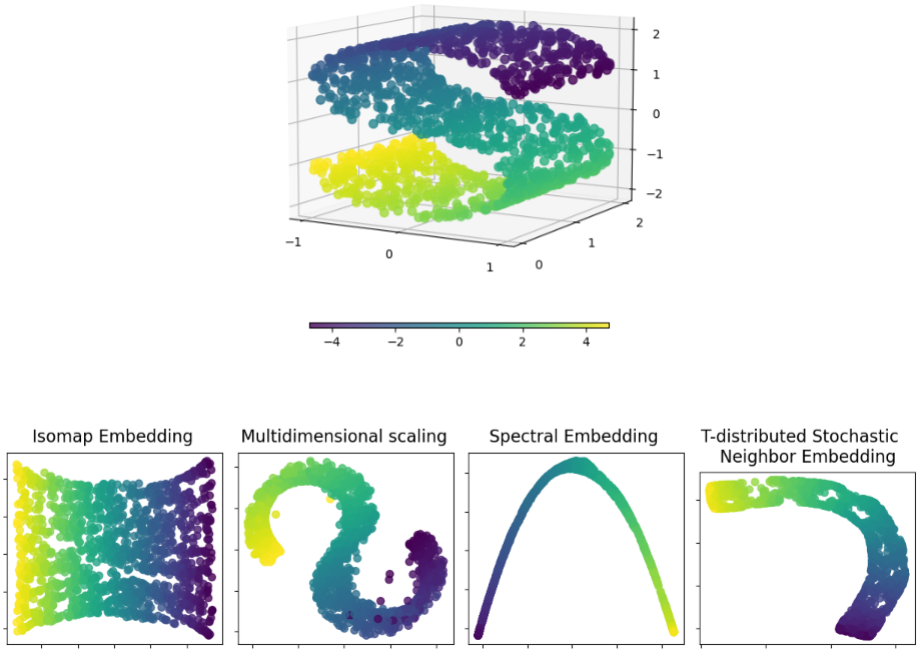


Figure 2.8: A visual comparison of Manifold Learning methods for an example of dimensionality reduction on a toy "S-curve" dataset in scikit-learn. "*Manifold Learning*", 2007-2023, av Scikit-Learn. (https://scikit-learn.org/stable/modules/manifold.html)

distribution Q, as defined in equation 2.5.

$$\mathcal{L} = KL(P||Q) = \sum_{c=1}^{M} P_c \log \frac{P_c}{Q_c} \tag{2.5}$$

As a result, the DEC method makes it possible to iteratively refine the initial clusters by matching the "*soft assignments*" to the target distribution. In this sense, this method can be associated with Self-Supervised Learning (SSL) techniques that use the predicted labels of a classifier to train the model by itself, based on its own confidence predictions.

2.4 Summary

This section provided a general overview of the computational techniques used in the field of ecoacoustics with a particular attention to applications related to DL for classification, and deep embedding for clustering analysis. To this end, species identification and location was first presented as a useful

way to estimate the population density of one species relative to another, and the techniques used to detect and classify the acoustic units of animal species in soundscape recordings were reviewed. Moreover, the multi-class classification problem in the context of DL was introduced and the technical aspects of DNN architectures as well as the acoustic features and pre-processing techniques used in ecocoustics were presented. Furthermore, the techniques used in clustering analysis were detailed and the DEC method was introduced as an interesting way to improve the quality of data clustering in the framework of DL.

Chapter 3

Related Work

A recurring problem in ecoacoustic projects is the lack of large labeled datasets. As a result, it is difficult to capture the vocalizations of rare tropical bird species because few audio recordings are accessible for training a DL classification model. In this chapter, different learning strategies are introduced to overcome the problem of multi-class classification with few samples of data. More precisely, the Meta-Learning framework is introduced as a relevant solution to build DL classification models that can be quickly adjusted for new tasks. Moreover, the Unsupervised Meta-Learning (UML) framework is considered as an interesting way to automatically build learning tasks for Few-Shot Image Classification, by assigning pseudo-labels to samples from unlabeled datasets.

3.1 Meta-Learning for Few-Shot Classification

In the context of ecoacoustics with DL, several strategies have attempted to tackle problems related to the lack of data, including multi-task learning [43], semi-supervised learning [37], weakly-supervised learning [44] or Self-Supervised Learning (SSL) [45]. In this thesis, Meta-Learning algorithms are investigated for their capability to learn from other learning algorithms. Meta-Learning algorithms consist in learning to quickly adapt to the learning tasks of a prior model to be optimized for a set of novel tasks. In that sense, the basic idea is to be able to "*learn to learn*" [46]. Therefore, Meta-Learning algorithms provide a more general understanding of learning and allow to solve classification tasks by exposure to multiple similar classification tasks [47]. For this purpose, learning is usually performed in an episodic manner, where one episode corresponds to an N-way-K-shot task [9].

3.1.1 Episodic Learning

Meta-Learning algorithms are studied using the N-way-K-shot classification task, where N corresponds to the number of classes and K is the number of examples for each class, as illustrated in Figure 3.1. For example, the goal might be to try to discriminate a set of 5 classes (5-ways) with only 1 sample per class (1-shot). This then allows us to consider episodic learning for training models and gain experience from other similar problems. Episodic learning is different from conventional mini-batch training because of the introduction of episodic task sets. In each episodic task, the model learns to predict the classes of unlabeled data (i.e. *query set*) using very few labeled examples (i.e. *support set*) [48]. Nevertheless, recent work suggests that competitive results can be obtained from classical training with simple Cross-Entropy loss overall training classes, compared to the more sophisticated episodic methods [49, 50]. Therefore, it is becoming increasingly common to use a classical training process to train the backbone of a neural network. In this thesis, the objective is to evaluate to what extent episodic training compared to classical training can improve the performance of a Few-Shot Image Classification task. Furthermore, Meta-learning is usually

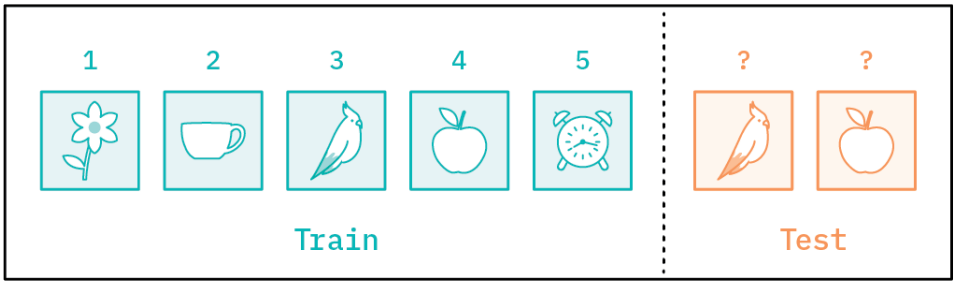


Figure 3.1: A Few-Shot Image Classification (5-ways-1-shot) task. "*Meta-Learning*", 2020, av Cloudera Fast Forward Labs. (https://meta-learning.fastforwardlabs.com/)

defined in two levels: meta-training and meta-testing, as illustrated in Figure 3.2. During the meta-training phase, several episodic tasks are defined. Here, this corresponds to classifying N-way bird species from K-shot spectrograms for each species. This then allows the knowledge learned through the episodic tasks to be reused to understand how the structure of the tasks varies across target domains and to classify new data in the meta-testing phase. Performance evaluation of the Meta-Learning algorithms consists of

using a set of test tasks where each of the classes is different from those used in the training tasks. This makes it possible to measure the ability of the model to correctly classify the *query set* based on its knowledge of the *support set*.

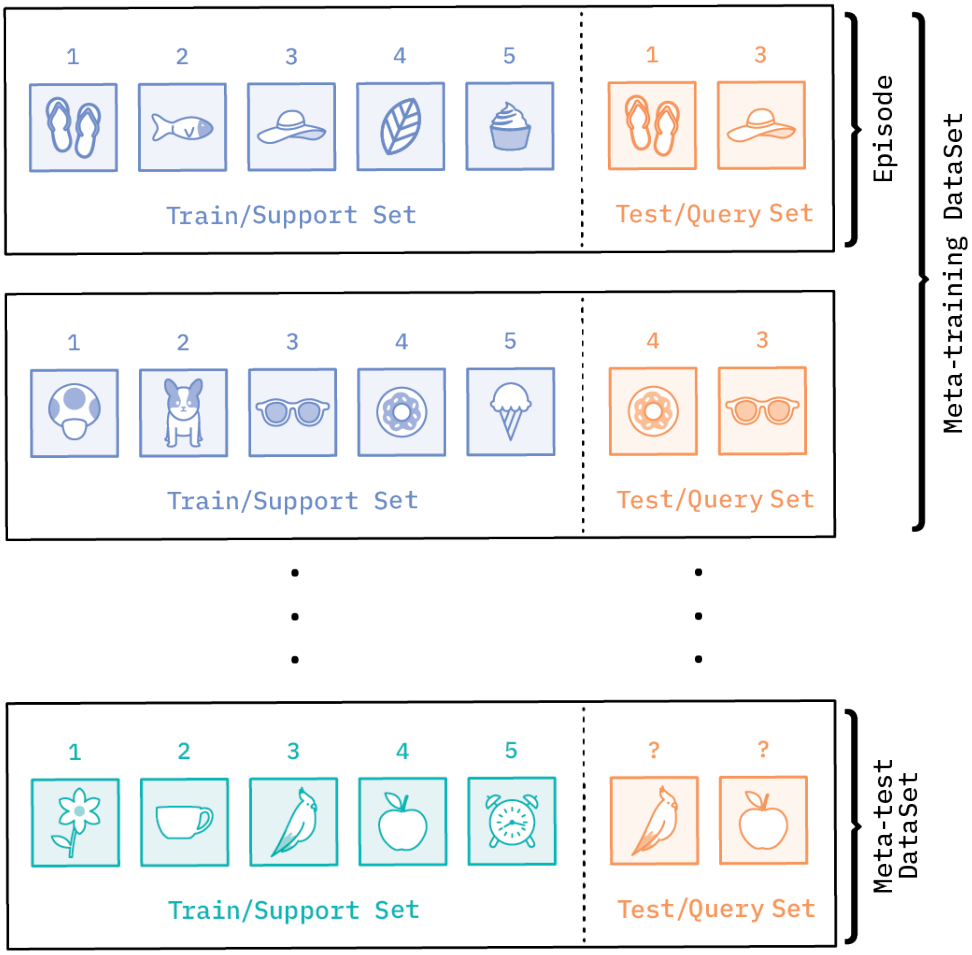


Figure 3.2: Meta-Learning data setup. "*Meta-Learning*", 2020, av Cloudera Fast Forward Labs. (https://meta-learning.fastforwardlabs.com/)

3.1.2 The Few-Shot Image Classification problem

One of the most popular examples of Meta-Learning is the Few-Shot Image Classification problem. Recently, the Meta-Learning framework has attracted increasing attention for the acoustic Few-Shot Image Classification

problem [51, 52]. The goal is to allow the classification of new images (e.g. spectrograms) from a handful of training examples. Some cases have only one example per class (i.e. One-Shot Learning) or none (i.e. Zero-Shot Learning). Moreover, Few-Shot Learning (FSL) has attracted increasing attention in the field of ecoacoustics with notably the introduction of the Task 5 of the DCASE Challenge 2021 (Few-Shot Bioacoustic Event Detection). However, Task 5 deals with the actual detection of onsets and offsets of events whereas the FSL framework is here used as an audio tagging task. Regarding the DCASE Challenge 2022, the best overall F-score in the evaluation set reached the 60% level. Table 3.1 presents the validation and evaluation of F-score results per team, as mentioned in [53]. Given that a unique dataset

Team Name	Validation set: F-score %	Evaluation set: F-score % (95% CI)
Du NERCSLIP 2 [54]	74.4	60.22 (59.66-60.70)
Liu Surrey 2 [55]	50.03	48.52 (48.18-48.85)
Martinsson RISE 1 [56]	60	47.97 (47.48-48.40)
Hertkorn ZF 2 [57]	61.76	44.98 (44.44-45.42)
Liu BIT-SRCB 4 [58]	64.77	44.26 (43.85-44.62)

Table 3.1: Ranking of the top 5 F-score results per team on the DCASE Challenge 2022 Task 5 datasets. Systems are ordered by higher scoring rank on the evaluation set. From "*Few-shot bioacoustic event detection at the dcase 2022 challenge*" by I. Nolasco *et. al.* 2022, arXiv.

is created ofr this thesis and that the FSL framework is used as an audio tagging task, the evaluation results for the Few-Shot Image Classification task obtained on the *mini*ImageNet dataset are also presented to get a broader idea of the performances of the Meta-Learning algorithms. Table 3.2 lists the accuracy of three commonly used Meta-Learning algorithms as mentioned in [47].

Algorithm	1-shot	5-shots
Matching Networks [9]	43.56%	55.31%
Prototypical Networks [10]	49.42%	68.2%
Relation Networks [59]	50.44%	65.32%

Table 3.2: Accuracy of 5-ways Few-Shot Image Classification tasks on the *mini*ImageNet dataset. From "*A summary of approaches to few-shot learning*", by A. Parnami *et. al*, 2022, arXiv.

The motivation around the Meta-Learning framework is nevertheless the same as in detection and classification tasks because it lies essentially in the possibility to train ANNs with a reasonable performance despite the few training examples available, or to facilitate data labeling when the cost of the latter is high.

3.1.3 Metric Learning

Meta-Learning algorithms are generally labeled as either metric-learning based or gradient-based meta-learner. In this thesis, a particular emphasis is placed on metric-learning based algorithms to measure the distance between the feature vectors that are produced by the last layers of a pre-trained network. In metric learning, the feature vector represents a relatively low dimensional space in which high dimensional vectors can be translated. These vectors can then be placed in a coordinate system, also called dimensional space, to allow the interpretation of points in space. Thus, the closer the "points" are to each other, the more similar they will be considered. Previous metric-learning based algorithms focused on pairwise comparisons of embeddings to determine the membership of two examples of data to the same class or to different classes (e.g. Siamese Networks [60] or Triplets Networks [61]). In the context of ecoacoustics with DL, terrestrial and underwater projects have reported the ability of these models to train relatively well with small or imbalanced data sets [62, 37]. The advancement of research in the Meta-Learning framework has notably allowed the development of multi-class algorithms capable of assigning new examples to a class among several (e.g. Matching Networks [9] or Prototypical Networks [10]). All of these algorithms use metric-learning methods to automatically construct task-specific distance metrics that can be measured in various ways. The basis of many similarity and dissimilarity measures is the Euclidean distance. This distance is calculated between two vectors A and B as follows:

$$\text{Euclidean Distance} = |\mathbf{A} - \mathbf{B}| = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{3.1}$$

According to [10], the squared Euclidean distance can greatly improve results depending on the network used. The only difference with the Euclidean distance is that it does not take the square root.

$$\text{Squared Euclidean Distance} = |\mathbf{A} - \mathbf{B}| = \sum_{i=1}^{n} (A_i - B_i)^2 \tag{3.2}$$

Moreover, the similarity between two points can also be measured by calculating the cosine similarity which determines the angle between the vectors rather than the distance between their extremities.

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.3}$$

There are numerous other distance-based similarity measures (e.g. Manhattan distance, Minkowski distance, Cross-Correlation, Jaccard Similarity, etc.). However, concerning ecoacoustics with DL, it is important to note that such measures only approximately represent the perceptual similarity of the vocalization of an animal [5]. Moreover, the acoustic perception of an animal vocalization can often differ from one species to another. Therefore, the similarity measure between, for example, two individual acoustic units, can be strongly affected by the pre-processing applied to the time-frequency representation on the one hand, and the type of similarity measure used on the other. To this end, another solution is to use Deep Metric Learning strategies to learn discriminative features produced by an ANN and to use it as a non-linear operator producing a similarity score. This has been notably introduced with the Relation Networks [59] that we will introduce in more detail in section 4.2.1. The main characteristic of this network is that it can predict a relation score as a means of predicting the relationship/similarity of embeddings produced by a CNN.

3.2 Unsupervised Meta-Learning

Most Meta-Learning algorithms are evaluated under supervised Few-Shot Image Classification tasks which nevertheless require a large number of labeled data. To tackle this problem, recent approaches based on Unsupervised Meta-Learning (UML) have been explored to allow the creation of high-quality embeddings with pseudo-labeled training data. In contrast to supervised Meta-Learning, the UML framework is characterized by a learning procedure, without supervision, that is useful to solve a wide range of new human-specified tasks [63]. This has led to the development of a wide field of applications to define new types of representations needed for feature detection or classification from unlabeled data. The basic idea is that features are learned from the pseudo-labeled data by analyzing the relationship between points in the dataset. This learning can then be reused to classify/cluster

sets with few data. In this thesis, the UML framework is used to develop unsupervised embedding algorithms capable of improving the clustering quality of unlabeled data. This can be achieved using various unsupervised methods providing an alternative way to consider real-world problems, like exploiting the support and query sets involved in the Meta-Learning framework as pseudo-labeled data. To the best of our knowledge, such an approach has never been established before in the context of ecoacoustics with DL.

3.2.1 Clustering-Based Unsupervised Methods

The basic idea of the UML framework is to use unlabeled data for the meta-training phase. According to [64], there are two common unsupervised methods to build tasks from the unlabeled dataset:

1. Comparative Self-Supervised (CSS)-based methods (as shown in Figure 3.3 (c)) which use data augmentations to create image pairs that can be used to build training tasks. To this end, Khodadadeh *et al.* proposed the method UMTRA [65] to enable the creation of synthetic tasks in the meta-training phase, using random sampling and augmentation.

2. Clustering-based methods (as shown in Figure 3.3 (d)), which use the pseudo-labels of the clusters generated by a clustering algorithm to build training tasks. To this end, Hsu *et al.* proposed the method CACTUs [63] that allowed the development of efficient models from a few samples of data for various tasks.

In this thesis, the main focus is on clustering-based methods to use unlabeled data as multiple clusters defining the pseudo-labels of our images. More precisely, UML algorithms use the pseudo-labels defined by a standard clustering algorithm as supervision to update the weights of an ANN. However, the meta-training phase is often limited by label inconsistency and limited diversity in the training set which can affect the model performance. According to [66], this is because the unsupervised embedding algorithms are not suitable for the clustering task. For example, the algorithms used with the method CACTUs, such as InfoGAN [67], BiGAN [68], ACAI [69], or DeepCluster [70], were originally designed for the pre-training phase of the model only to refine the features extracted in the downstream tasks. Table 3.3 presents the baseline performances of unsupervised embedding algorithms trained from scratch on the *mini*ImageNet dataset. The best overall accuracy in the evaluation set reached the 50% level with the UFLST method [48]. The results are averaged over 1000 downstream tasks.

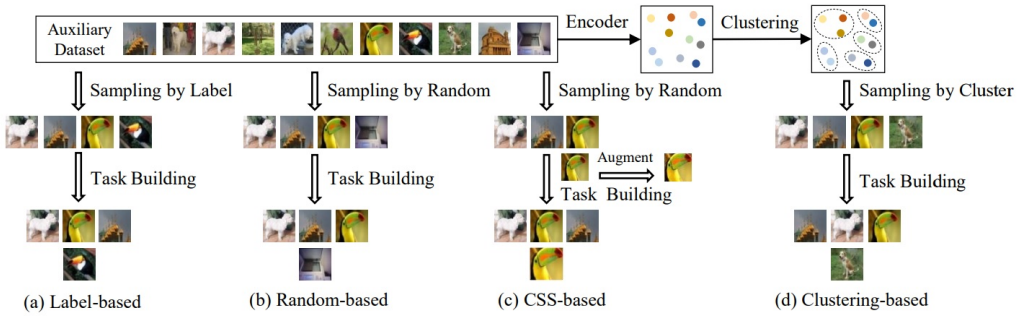


Figure 3.3: The four baselines from the view of sampling. (a) Label-based baseline, which is a supervised baseline. Since the images have category labels, two of the four sampled images belong to the same category. (b) Random-based baseline, in which four images are randomly sampled, and the label of tasks is randomly determined. (c) CSS-based baseline, in which three images are randomly sampled, and then one of the images is selected to obtain another view through data augmentation. (d) Clustering-based baseline, in which first all images are divided into multiple clusters by a clustering algorithm, and then four images are selected with cluster ids as labels. From "*Unsupervised few-shot image classification by learning features into clustering space*", by S. Li *et. al*, Conference, Tel Aviv, Israel, October 23, 2022, Springer.

Algorithm	1-shot	5-shots
BiGAN *k-Nearest Neighbors* [68]	25.56%	31.10%
BiGAN *Linear Classifier* [68]	27.08%	33.91%
DeepCluster *k-Nearest Neighbors* [70]	28.90%	42.25%
DeepCluster *Linear Classifier* [70]	29.44%	39.79%
InfoGAN [67]	29.81%	36.47%
UFLST [48]	37.75%	50.95%

Table 3.3: Baseline performances for 5-ways Few-Shot Classification tasks with models trained from scratch on pseudo-labeled data. The data is pseudo-labeled using a clustering algorithm on the *mini*ImageNet dataset. From "*Unsupervised few-shot learning via self-supervised training*", by Z. Ji *et. al*, 2019, arXiv.

3.3 Summary

This section provided an overview of the work related to a recurring problem in ecoacoustic projects, namely: the lack of large labeled datasets. For

this purpose, the Meta-Learning framework was reviewed, with a particular emphasis on metric-learning based algorithms to introduce the concept of episodic learning as well as the Few-Shot Image Classification problem. Moreover, the Unsupervised Meta-Learning (UML) framework was introduced as an interesting alternative to tackle the problems related to the lack of large labeled datasets in the context of ecoacoustics with DL. Specifically, the clustering-based methods have been identified as a relevant solution to train Meta-Learning algorithms using pseudo-labeled data. This makes it possible to improve the performance of a classifier even though there is a lack of labeled data available. Finally, although the use of unsupervised methods in the UML framework is generally considered to tackle problems related to Few-Shot Image Classification tasks, the following sections will evaluate their ability to improve the quality of the clustering of unlabeled data.

Chapter 4

Material & Methods

In this section, our proposed framework is introduced in order to classify rare tropical bird species with limited annotated data available and facilitate the work of ecoacousticians for the management of acoustic data and the identification of potential new taxa. In the context of ecoacoustics, more and more projects are interested in studying animal populations in developing countries or in locations/regions that are difficult to access. However, most of the data present in current bird databases is mainly concentrated in the United-States and Western Europe [6]. Therefore, the development of an ecoacoustic project outside these regions often requires the contribution of qualified field experts to label data that is sometimes unknown. For this purpose, material and methods are presented to facilitate the identification of rare bird species in tropical environments. This includes the creation of a unique dataset, composed of acoustic units of nocturnal and crepuscular bird species living in the American tropics collected and segmented from the Xeno-Canto database[1]. This serves as a basis to (i) define an efficient method for the Few-Shot Image Classification problem by comparing and evaluating different pre-existing methods allowing us to tackle the problems related to the lack of data, and (ii) review a method aiming at gradually improving the quality of the clustering using an iterative learning process, in order to facilitate the labeling of unlabeled data.

4.1 Darksound Dataset

The Darksound dataset is built as an open-source and code-based dataset for the evaluation of Meta-Learning algorithms in the context of ecoacoustics with DL. The dataset is easily accessible and downloadable with the

[1] https://xeno-canto.org/

following link: https://www.kaggle.com/datasets/joachipo/darksound. The particularity of this dataset is that it is composed of acoustic units, also called Regions of Interest (ROIs), of 290 nocturnal and crepuscular bird species living in tropical environments. All the ROIs in the Darksound data set have a sampling rate of 48 kHz and are faded in and out to avoid aliasing effects due to window effects. Moreover, each ROI is padded to a maximum duration of 3 seconds to obtain input images of equal size for training the model.

4.1.1 Data Acquisition

The data used for this work include soundscape recordings collected and segmented from the Xeno-Canto database, a collaborative project dedicated to sharing bird sounds around the world. More specifically, the Xeno Canto web Application Programming Interface (API v2)[2] is used to build the dataset. According to the API documentation, the data can be used without restriction, with a limit of 10 queries per second. The data is accessed by sending query parameters that return a JSON object containing details about the records found with the given query. An explanatory notebook presenting the different steps necessary to acquire the data in tropical environments is available on this link: https://github.com/joachimpoutaraud/darksound/blob/master/notebooks/01-builing_dataset.ipynb. This can be easily re-appropriated for the creation of new data sets involving new environments.

Bambird

Data acquisition was facilitated by the use of the Bambird package[3] developed with the Python programming language by the EcoAcoustics Research (EAR)[4] team of the Muséum National d'Histoire Naturelle (MNHN) in Paris. This package allows the use of a data-centric function that automatically segments audio recordings before assigning a pseudo-label to each unsupervised segmented audio sample [71]. the function used is defined in three parts, namely:

1. Time-frequency acoustic unit segmentation
2. Feature computation for each acoustic unit

[2] https://xeno-canto.org/explore/api
[3] https://github.com/ear-team/bambird
[4] https://ear.cnrs.fr

3. Unsupervised classification of each acoustic unit as bird song or noise with a clustering algorithm

More precisely, the first step consists in segmenting the data from a spectrogram by delimiting Regions Of Interest (ROIs) around the salient sounds. The delimitation of the ROIs is performed using a regional growth segmentation method known as binarization by hysteresis thresholds [72]. Then, vectors of features are extracted for each ROI by convolving the ROIs with a series of 2D Gabor filters to extract the spectro-temporal characteristics according to the selected resolution [18, 71]. This allows the clustering of the extracted features for each bird species entered in the Xeno-Canto query, using the DBSCAN [73] clustering algorithm. The biggest cluster of ROIs is finally selected for each species and represents the pseudo-labels of the species in question. An overview of the complete labeling function design process is illustrated in Figure 4.1. Results corresponding to the evaluation of the Bambird workflow are presented in Appendix A.

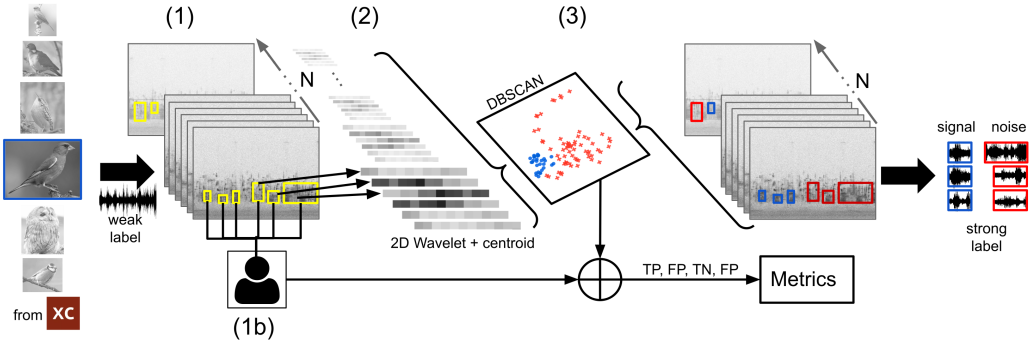


Figure 4.1: Bambird workflow: after collecting N weakly labelled audio recordings of a focal species from the Xeno-Canto database, labeling function workflow consists of (1) segmenting each audio recording into time-frequency acoustic units, (2) calculating 49 features (2D wavelet coefficients ($n = 48$) + frequency centroid ($n = 1$)) of each acoustic unit, (3) pseudo-labeling of all acoustic units into signal (bird song belonging to the focal species) or noise (everything else) with DBSCAN. In parallel, an expert can annotate all acoustic units (1b) to calculate the metric to evaluate the performance of the labeling function. From "*Unsupervised classification to improve the quality of a bird song recording dataset*", by Michaud *et. al*, Ecological Informatics, 2023, Elsevier.

Training set

The training set included 249 species of tropical birds vocalizing at night, which were automatically chosen. This included an initial number of 2,638 audio recordings requested from the Xeno-Canto database. The audio recordings were requested according to five parameters: (i) the audio quality, which was strictly greater than C with quality ratings ranging from A (highest quality) to E (lowest quality) in the advanced queries parameters of the Xeno-Canto database, (ii) the duration, which corresponded to a maximum of 120 seconds, (iii) the time of day at which bird species were recorded to download only species vocalizing at night, and (iv) the geographic coordinates that surrounded the Equator in America, so that the difference between daytime hours and night-time hours remained fairly constant throughout the year. The geographical coordinates were defined according to the latitude of the Tropics, with the Tropic of Cancer in the Northern Hemisphere at 23°26'10.6"N and the Tropic of Capricorn in the Southern Hemisphere at 23°26'10.6"S. Average night-time hours were established according to the geographical coordinates of the Tropics in America (minimum latitude: -23.439, minimum longitude: -92.734, maximum latitude: 23.439, maximum longitude: -34.789). This made it possible to define the average sunrise (04:00) and sunset (19:00) times of the year by using the Norwegian website *timeanddate*[5] which allowed one to retrieve time and time zone information from anywhere in the world. An overview of the geographical distribution of all the recordings in the Darksound data set is presented in Figure 4.2. The segmentation of the audio recordings was performed using the Bambird package with a frequency band between 250 and 2500 Hz corresponding to the frequency bands of the target species to obtain a number N of ROIs from which were features extracted. Particular attention was taken to removing species with less than 10 ROIs, as well as ensuring that no target species from the validation and test sets were found in the training set. Finally, a clustering of the extracted features was performed for each species to define several clusters on which the biggest one has been kept. As a result, each query species of the training set was represented by the number of ROIs found in the biggest cluster of its class. This resulted in the recovery of 249 species with a total number of 4,149 pseudo-labeled ROIs for the training set. Regarding the species requested for the training set, it is important to note that some of them could *potentially* be associated with other bird species or groups of animals (e.g. anurans or insects) if it turned out that the majority of the ROIs collected during the segmentation phase corresponded to a different animal species than the one specified in the request. In this

[5] https://www.timeanddate.com/

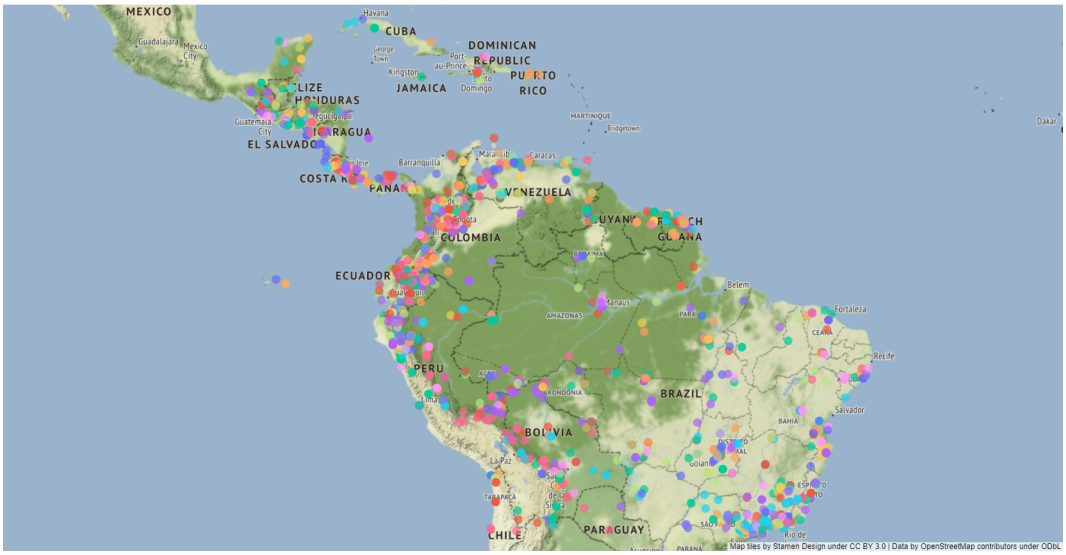


Figure 4.2: Geographical distribution of the Darksound training set in the Tropics. The training set includes 249 species of tropical birds vocalizing at night for an initial number of 2,638 audio recordings requested from the Xeno-Canto database.

sense, the content of each class in the training set depended solely on the pseudo-labeling of the clustering algorithm corresponding to step 3 of the Bambird package. Meaning that, the training set was not labeled in a supervised manner.

Data oversampling was performed for each class of the training set which had a total number of ROIs less than 50. This is so that each class had a number of ROIs equal to 50. This avoided problems related to data imbalance and enhanced the training phase of the model. To do this, we artificially augmented the data using the Python package *audiomentations*[6] and applied the following waveform transformations:

1. AddGaussianSNR (add Gaussian noise to the input)
2. AirAbsorption (a lowpass-like filterbank with variable octave attenuation that simulates attenuation of high frequencies due to air absorption)
3. Time-Stretch (change the speed or duration of the signal without changing the pitch)

[6] https://iver56.github.io/audiomentations/

4. Pitch-Shift (pitch shift the sound up or down without changing the tempo)

5. Trim (trim leading and trailing silence from an audio signal)

Regarding the Pitch-Shift transformation, special attention was paid to the average values of the spectral centroid of each class. Specifically, the minimum and maximum average spectral centroid values were calculated on the biggest cluster of each class to define the range of values allowed for the pitch transformation. This ensures that the meaning of the data elements was not changed. Regarding the Time-Stretch transformation, the rate of change of the total duration of the signal was changed by 25% to slow down or speed up the audio without changing the pitch. For the rest of the transformations, the default parameters of the package were used.

Validation and Test Sets

The validation and test sets includes the vocalizations (i.e. ROIs) of 41 other nocturnal and crepuscular tropical bird species that were manually selected. The different species with their associated number of ROIs are presented in Table 4.1. This includes a total number of 1,242 ROIs with 20 classes and 618 ROIs for the validation set, and 21 classes and 632 ROIs for the test set. The target species include nocturnal raptors and tinamous, respectively "umbrella" and "sentinel" species of the Amazonian forest. The selection is made on the whole American continent to retrieve a maximum of data given their rarity. Figure 4.3 shows a bar chart with the number of ROIs obtained per target species.

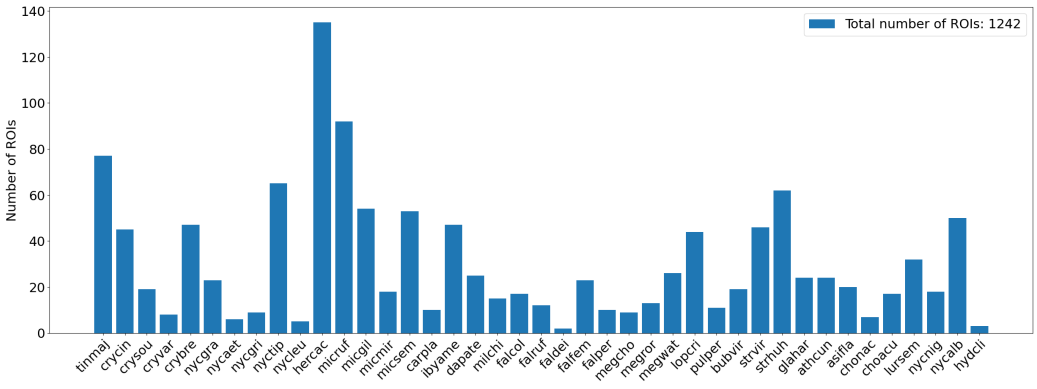


Figure 4.3: Total number of the nocturnal and crepuscular species in the validation and test sets before augmentation. The corresponding abbreviations of the species are filled in Table 4.1.

Species	Abbreviation	Family	Number of ROIs
Hydropsalis climacocerca	HYDCLI	Caprimulgidae	3
Lurocalis semitorquatus	LUCSEM	Caprimulgidae	32
Nyctidromus albicollis	NYCALB	Caprimulgidae	50
Nyctipolus nigrescens	NYCNIG	Caprimulgidae	18
Chordeiles acutipennis	CHOACU	Caprimulgidae	17
Chordeiles nacunda	CHONAC	Caprimulgidae	7
Nyctiprogne leucopyga	NYCLEU	Caprimulgidae	65
Daptrius ater	DAPATE	Falconidae	25
Falco femoralis	FALFEM	Falconidae	23
Falco rufigularis	FALRUF	Falconidae	12
Ibycter americanus	IBYAME	Falconidae	47
Micrastur ruficollis	MICRUF	Falconidae	92
Micrastur semitorquatus	MICSEM	Falconidae	53
Milvago chimachima	MILCHI	Falconidae	15
Caracara plancus	CARPLA	Falconidae	10
Falco columbarius	FALCOL	Falconidae	17
Falco deiroleucus	FALDEI	Falconidae	2
Falco peregrinus	FALPER	Falconidae	10
Herpetotheres cachinnans	HERCAC	Falconidae	135
Micrastur gilvicollis	MICGIL	Falconidae	54
Micrastur mirandollei	MICMIR	Falconidae	18
Nyctibius aethereus	NYCAET	Nyctibiidae	6
Nyctibius grandis	NYCGRA	Nyctibiidae	23
Nyctibius griseus	NYCGRI	Nyctibiidae	9
Nyctibius leucopterus	NYCLEU	Nyctibiidae	5
Asio flammeus	ASIFLA	Strigidae	20
Glaucidium hardyi	GLAHAR	Strigidae	24
Megascops roraimae	MEGROR	Strigidae	13
Pulsatrix perspicillata	PULPER	Strigidae	11
Strix huhula	STRHUH	Strigidae	62
Strix virgata	STRVIR	Strigidae	46
Athene cunicularia	ATHCUN	Strigidae	24
Bubo virginianus	BUBVIR	Strigidae	19
Lophostrix cristata	LOPCRI	Strigidae	44
Megascops choliba	MEGCHO	Strigidae	9
Megascops watsonii	MEGWAT	Strigidae	26
Crypturellus cinereus	CRYCIN	Tinamidae	45
Crypturellus soui	CRYSOU	Tinamidae	19
Crypturellus variegatus	CRYVAR	Tinamidae	8
Crypturellus brevirostris	CRYBRE	Tinamidae	47
Tinamus major	TINMAJ	Tinamidae	77

Table 4.1: Number of Region Of Interests (ROIs) of nocturnal and crepuscular tropical bird species present in the validation and test sets of the Darksound data set. This includes five bird families with a total number of 1,242 ROIs with 20 classes and 618 ROIs for the validation set, and 21 classes and 632 ROIs for the test set.

All the ROIs of the validation and test sets have been labeled by a member of the EAR team from the MNHN. Some of the classes had a number of ROIs lower than 6 (*Hydropsalis climacocerca*, *Falco deiroleucus*, and *Nyctibius leucopterus*). Given that the performance of the Meta-Learning algorithms is usually evaluated in two configurations (i.e. N-way-1-shot and N-way-5-shots), classes with less than 6 ROIs were artificially augmented to

a minimum of 6 ROIs per class (i.e. 5 shots + 1 query) to allow the comparison of the performances of the algorithms. Oversampling of the data was performed in the same manner as on the training set.

4.1.2 Data Preprocessing

Commonly, CNN such as ResNet18 performs image processing on multi-channel images, where each channel represents a color and each pixel consists of three channels (usually RGB). In this thesis, a multi-channel "sound image" is proposed as input data to represent how the sound energy of animal acoustic units is communicated and manifested over time. More precisely, an input spectrogram is decomposed in Harmonic and Percussive components, and its Derivative is calculated to obtain a 3-channels (HPD) "sound image" as illustrated in Figure 4.4. The HPD image is useful for representing the sound energy characteristics of animal acoustic units and defining meaningful features for the classification phase.

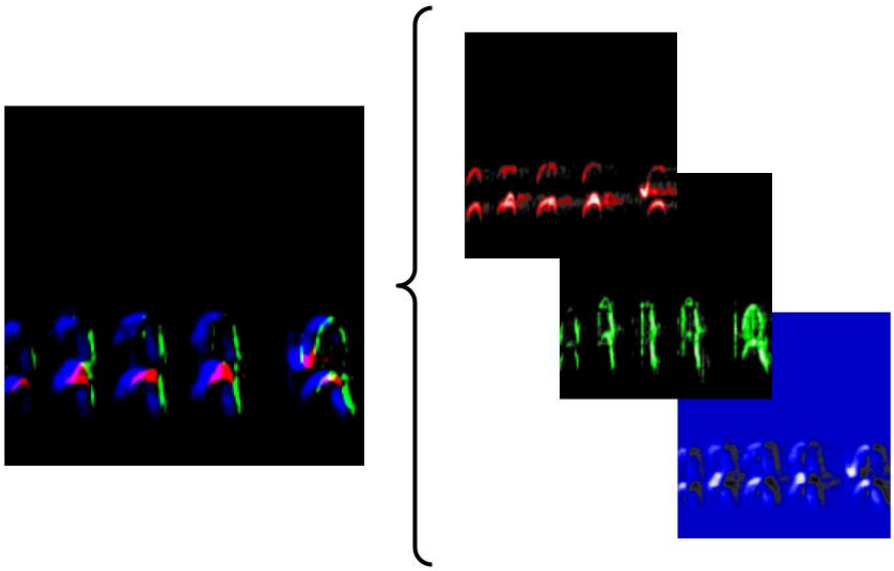


Figure 4.4: Schematic representation of a Harmonic-Percussive-Derivative (HPD) sound image. Left: the HPD image. Right: Harmonic component (top), Percussive component (middle), and Derivative of the original spectrogram in dB (bottom).

Harmonic-Percussive Source Separation (HPSS)

The decomposition of a spectrogram into its harmonic component and percussive component has attracted much interest in related literature and can be applied as a pre-processing step for DL classification tasks [74]. In particular, its use has been extended with the separation of the residual component which allows to refine the separation of the harmonic and percussive components [75]. This approach can be useful for distinguishing certain animal vocalizations since a different species may have a more expressive call in one of its components [76]. Thus, using this approach as a pre-processing step on a spectrogram may allow the development of more expressive CNN integration for the classification of animal vocalizations. For example, the use of 3-channels spectrograms as input to a CNN (treated in the same way as the channels of an RGB image), can allow the establishment of latent space representations more representative of the type of sound components conveyed by an animal vocalization. As a result, by applying Harmonic and Percussive Source Separation (HPSS) with median filtering on original spectrograms in dB [74], it is possible to isolate the Harmonic and Percussive components of animal vocalizations to improve their visual representation.

Delta Features

Acoustic signals produced by animals can be described as a sequence of transitions between acoustic units. A common method for extracting information about these transitions is to determine the first difference in the signal characteristics, known as the *delta* of a feature. Delta features are commonly used in machine learning because they are easy to compute and provide a clear advantage over instantaneous features. In this work, the *delta* of an input spectrogram s_k in dB is computed at time instant k, with the following equation:

$$\Delta_k = s_k - s_{k-1}. \tag{4.1}$$

4.2 Experimental Design

This section presents the experiments conducted in this thesis and the evaluation criteria that are used when evaluating the models. Specifically, this section focuses on the evaluation of Meta-Learning algorithms that are compared and used to generate useful latent feature representations for data clustering. For this purpose, four types of experiments are performed, namely:

1. Comparison of the raw performance of three Meta-Learning algorithms on the Darksound dataset.

2. Comparison of episodic training against classical training for fine-tuning Meta-Learning algorithms to determine the best training method for Few-Shot Image Classification tasks.

3. Optimization of Meta-Learning algorithms and evaluation of their ability to classify classes that were not seen during training.

4. Extraction of Meta-embeddings to cluster the data in the latent space, and refine the clusters by iteratively fine-tuning Meta-Learning algorithms.

4.2.1 Meta-Learning Algorithms

As mentioned earlier, Meta-Learning algorithms are generally labeled as either metric-learning based or gradient-based meta-learner. In this thesis, specific emphasis is placed on metric-learning based algorithms that are used for performing the experiments. Architectures of the metric-learning based algorithms have been adapted from the *EasyFSL*[7] package developed by Etienne Bennequin and implemented using the Python library Pytorch[8], a Torch-based machine learning framework. The code of the model architectures is available at the following address: https://github.com/joachimpoutaraud/darksound. The implementation is based on the comparison of three commonly used Meta-Learning algorithms for the Few-Shot Image Classification of ROIs that have been segmented using the Bambird package. For each model, Transfer Learning (TL) is used to benefit from the learning of a pre-trained model on image classification. This allows us to tackle the problem related to the large amount of training data that Deep Neural Networks (DNN) usually require to achieve satisfactory performance. Specifically, a ResNet18 model pre-trained on the ImageNet database is used as a backbone. The ImageNet database contains 1,000 object classes with 1281,167 training images, 50,000 validation images, and 100,000 test images [77]. Although this database does not contain spectrograms, its use in the context of ecoacoustics with DL is common and has allowed the learning of a variety of image features useful for spectrogram classification [78, 79]. Fully Connected (FC) layers of the ResNet18 are removed from the model implementation and replaced with a Meta-Learning algorithm. Few-Shot Image Classification is then performed by learning a measure on the Darksound dataset. To

[7] https://github.com/sicara/easy-few-shot-learning
[8] https://pytorch.org/

do this, the *query* images of a new class are classified based on the learning of a measure that computes distances to the *support* images. In DL, the distance measure and the feature integration are often learned separately to isolate as much of the non-linear structure of the data as possible. According to [80], this allows for the generation of more discriminative feature representations. For each model, the last layer is composed of several nodes corresponding to the number of target species (N-way). The probability distribution is then calculated using the logarithm of the Softmax activation function which allows for strongly penalizing the model when it fails to predict a correct class. As a result, the model produces a vector of N-way scores, where the value closest to 1 corresponds to the species predicted by the model. In the following, details of the Meta "metric-learning" algorithms are introduced. This includes algorithms that are chosen based on the aspect they improve, namely: 1) learning feature embeddings, 2) learning class representations, and 3) learning distance or similarity measures.

Matching Networks: Learning Feature Embeddings

Matching Networks are considered the first metric learning algorithm designed to solve Few-Shot Image Classification problems [9]. Their operation is based on methods of learning to integrate high-dimensional features into a low-dimensional space so that discriminative features can be extracted to perform a generalized form of nearest neighbor classification. According to [81], the label of the one-shot coded *query set* $\hat{\mathbf{y}}$ is defined as the weighted sum of all labels in the one-shot coded *support set* $\{\mathbf{y}_{nk}\}_{n,k=1}^{NK}$:

$$\hat{\mathbf{y}} = \sum_{n=1}^{N} \sum_{k=1}^{K} d[\mathbf{x}_{nk}, \hat{\mathbf{x}}] \mathbf{y}_{nk} \tag{4.2}$$

To calculate the similarity $d[\mathbf{x}_{nk}, \hat{\mathbf{x}}]$, each example from the support set $\mathbf{x}_{nk}$ goes through a CNN $\mathrm{f}[\bullet]$ that produces a latent space representation, and each example from the query set $\hat{\mathbf{x}}$ goes through another CNN $\mathrm{g}[\bullet]$ that produces another representation. A schematic representation of the architecture of the Matching Networks is illustrated in Figure 4.5. Cosine similarity is then calculated between the different latent space representations with the following equation:

$$d[\mathbf{x}_{nk}, \hat{\mathbf{x}}] = \frac{\mathrm{f}[\mathbf{x}_{nk}]^T \ \mathrm{g}[\hat{\mathbf{x}}]}{|| \ \mathrm{f}[\mathbf{x}_{nk}]|| \cdot || \ \mathrm{g}[\hat{\mathbf{x}}]||}, \tag{4.3}$$

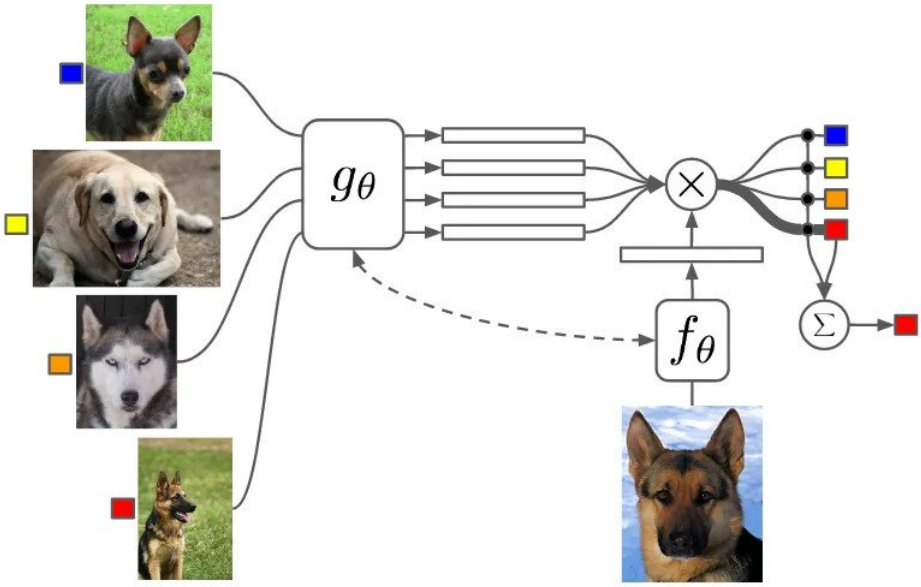


Figure 4.5: Matching Networks architecture. From "*Matching Networks for One Shot Learning*", by O. Vinyals *et. al*, 2017, arXiv

The result is normalized with the logarithm of the Softmax function to produce positive similarities whose sum is equal to one.

$$a[\hat{\mathbf{x}}_{nk}, \mathbf{x}] = \frac{\exp[d[\mathbf{x}_{nk}, \hat{\mathbf{x}}]]}{\sum_{n=1}^{N} \sum_{k=1}^{K} \exp[d[\mathbf{x}_{nk}, \hat{\mathbf{x}}]]} \tag{4.4}$$

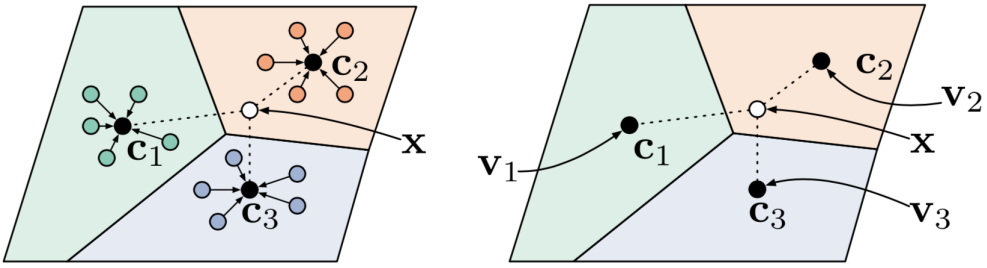
That way, the model is trained in an end-to-end fashion by calculating the Cross-Entropy loss on the actual labels and the predicted labels. The loss is finally back-propagated through CNN so that it can learn from its errors. Nevertheless, the disadvantage of Matching Networks is that they are not robust to data imbalance [81]. As a result, the more support examples there are for some classes, the more classes with frequent training data may dominate.

Prototypical Networks: Learning Class Representations

To overcome the problem of data imbalance between classes, Prototypical Networks were introduced in [10]. These networks use class *prototypes* that serve as reference vectors for each class $c \in \mathcal{C}$. The vectors $\mathbf{v}_c$ are thus constructed by taking the simple or weighted average of the latent space

representations from the examples of the class, as illustrated in Figure 4.6.

$$\mathbf{v}_c = \frac{1}{|S_c|} \sum_{(\mathbf{x}_i, y_i) \in S_c} f_\theta(\mathbf{x}_i) \tag{4.5}$$



(a) Few-shot (b) Zero-shot

Figure 4.6: Prototypical networks in the Few-shot and Zero-shot scenarios. Left: Few-shot prototypes c_k represents the mean of embedded support examples for each class. Right: Zero-shot prototypes c_k are produced by embedding class metadata v_k. From "*Prototypical networks for few-shot learning*", by J. Snell *et. al*, 2017, Advances in Neural Information Processing Systems 30.

Prototypical Networks thus allow to learn the latent representation or the *prototype* of each class using episodic training to minimize the Cross-Entropy loss. To do this, the similarity is computed at each episode as the negative multiple of the squared Euclidean distance between each prototype and the query embedding. Furthermore, [10] mentions that the higher the number of classes in the support set, the better the performance.

Relation Networks: Learning Distance/Similarity Measures

Unlike Matching Networks and Prototypical Networks, which both use a distance metric defined in advance to compare the latent space representations produced by the CNN, Relation Networks [59] learn their own distance metric to predict the relationship/similarity of embeddings using a CNN classifier g_ϕ, as illustrated in Figure 4.7. Apart from that, the approach is quite similar to that of Prototypical Networks since the simple or weighted average of the embeddings is performed for each class of the support set to form a prototype and thus alleviate data imbalance problems. The relationship score between a pair of inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, is calculated as follows $r_{ij} = g_\phi([\mathbf{x}_i, \mathbf{x}_j])$ where $[.,.]$

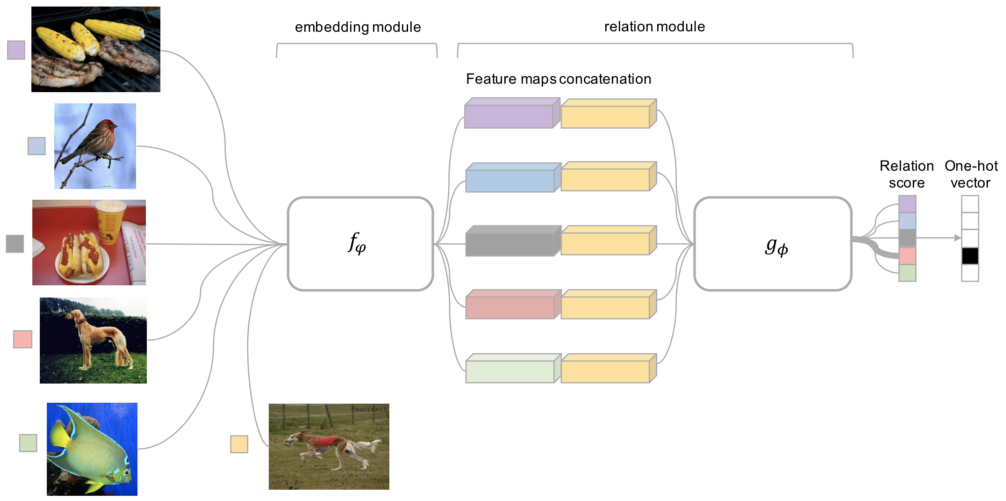


Figure 4.7: Relation Network architecture for a 5-way 1-shot problem with one query example. From "*Learning to Compare: Relation Network for Few-Shot Learning*", by F. Sung *et. al*, 2018, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

represents the concatenation of each embedding with the query embedding. A relationship score is finally obtained between 0 and 1. Predictions are obtained by comparing the relation scores with the query labels that are encoded as one-hot vectors, where 1 indicates that the query example belongs to this class prototype.

4.2.2 Meta Embedded Clustering

Recent works in the UML framework used clustering-based unsupervised methods as an interesting way to train a model on pseudo-labeled data [63, 48, 66]. According to [48], clustering-based unsupervised methods can improve the organization of data points for the model to discover "*the underlying structure of data gradually*". Based on this observation, the Meta Embedded Clustering (MEC) method is proposed as an alternative to the DEC method introduced in subsection 2.3.2. MEC method is performed on the Darksound dataset to refine the clusters of the 21 target species that are present in the test set. The goal is to determine the final number of clusters in an unsupervised way to facilitate the identification and visualization of rare tropical bird species in unlabeled datasets. The MEC method is organized in two phases with parameter initialization of the model, and

parameter optimization for clustering the data using KL divergence loss (see equation 2.5). That way, it is possible to improve the quality of data clustering on unlabeled data to evaluate the capacity of the method to learn "*the underlying structure of data gradually*" and determine the number of clusters. For this purpose, (1) the data is passed through the initialized model to (2) get an initial estimate of the non-linear mappings and avoid the "*curse of dimensionality*" [42]. Then, (3) the clustering algorithm is performed on the latent space to (4) build a pseudo-labeled dataset. Eventually, (5) the model is fine-tuned on the pseudo-labeled dataset for n episodic tasks. This process is repeated for 20 iterations to refine the initial clusters. An illustration of the MEC method is presented in Figure 4.8.

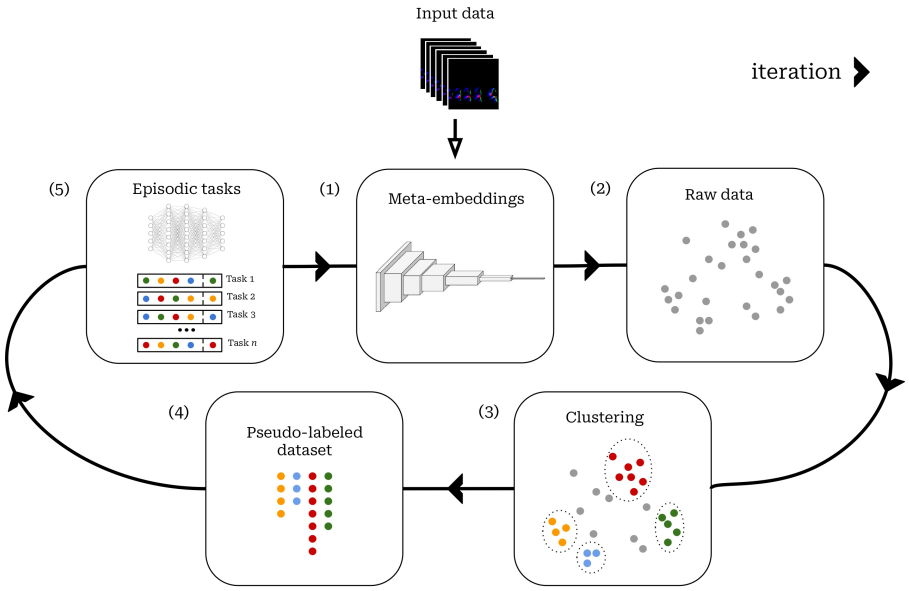


Figure 4.8: Meta Embedded Clustering (MEC) method. (1) Data is passed through the initialized model. (2) Initial estimate of the non-linear mappings are computed to avoid the curse of dimensionality. (3) Clustering algorithm is performed on the latent space. (4) Pseudo-labeled dataset is built from the clustering algorithm's predictions. (5) Model is fine-tuned on the pseudo-labeled dataset for n episodic tasks.

4.2.3 Experiments

Comparison of the raw performance of three Meta-Learning algorithms on the Darksound data set

For this first experiment, the performance of the Meta-Learning algorithms is evaluated and compared with input images (i.e. spectrograms) that have been pre-processed in different ways (i.e. with 3-channels (HPD) "sound images"). To do this, a pre-trained feature extractor (ResNet18) is used. Fully Connected (FC) layers of the network are removed and replaced by a Meta-Learning algorithm placed on the network top. When using the Matching and the Prototypical Networks, the probability distribution is computed using the logarithm of the Softmax activation function, and the Cross-Entropy loss is used to measure the distance from the ground truth values (see equation 2.3). The goal is here to minimize the loss to optimize the model, where a perfect model has a Cross-Entropy loss of 0. Regarding the Relation Networks, the Mean Squared Error (MSE) loss function is used with no activation function since the CNN focuses on predicting relationship scores which is more like a regression than a classification problem [11]. MSE loss function is defined in equation 4.6, where x and y are D dimensional vectors, and x_i denotes the value on the ith dimension of x.

$$\mathcal{L}_{MSE} = \sum_{i=1}^{D}(x_i - y_i)^2 \tag{4.6}$$

Adam optimizer [82] is chosen for adjusting the weights of the models based on the moving average of gradients calculated in the current and previous epochs. This optimizer is commonly used in DL for multi-class classification tasks. Implementation of the Adam optimization method is performed with a default learning rate of 0.0001 and weight decay of 0. Preliminary training on the Darksound dataset is done using the three aforementioned models to select the best-performing model. Each model is trained for 20 epochs on various N-way-K-shot tasks, where $N = \{5, 20\}$ and $K = \{1, 5\}$. A scheduler is used for reducing the learning rate by a default factor of 0.1 when an indicator has stopped improving after 5 epochs. In addition, an early stop is applied if the performance of the model has not improved after 10 epochs.

Evaluation of the performances of the model is performed using the confusion matrix to produce four observations with (i) the True Positive (**TP**), indicating that the observation is predicted to be positive and the prediction is true, (ii) the True Negative (**TN**) indicating that the observation is predicted to be negative and the prediction is true, (iii) the False Positive (**FP**),

indicating that the observation is predicted to be positive, but the prediction is false, and (iv) the False Negative (**FN**), indicating that the observation is predicted to be negative, but the prediction is false. These observations are then used to calculate the accuracy of the classification model as defined in equation 4.7.

Performance metrics

Accuracy describes the performance of a model on positive and negative samples in a symmetric way. It measures the rate of correct predictions with the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.7}$$

Comparison of episodic training against classical training for fine-tuning Meta-Learning algorithms to determine the best training method for Few-Shot Image Classification tasks

For this second experiment, comparison of episodic training against classical training is performed for Few-Shot Image Classification tasks. Classical training of models requires the set of the batch size that can have a decisive influence on the results [83]. Thus, for a fair comparison in the case of classical training, the batch size is determined according to the episodic training parameters. For example, if a model is trained on an N-way-K-shot task, the batch size is determined according to the number of N-ways times the total number of images per class $K+Q$, where K corresponds to the number of support samples per class and Q to the number of query samples per class. As a result, the batch size for a 5-ways-5-shots task with 1 query sample per class is equal to $5 \times (5 + 1) = 30$. For each comparison, experiments are conducted with a model fine-tuned for 20 epochs on the Darksound training set. The model is optimized using the Adam optimizer initialized with a default learning rate of 0.0001 and weight decay of 0. A scheduler is reducing the learning rate by a default factor of 0.1 when the training loss stopped improving after 5 epochs. Moreover, an early stop is applied if the validation accuracy of the model has not improved after 10 epochs. The architecture of the model is composed of a ResNet18 pre-trained on ImageNet from which the FC layers have been removed and replaced with a Meta-Learning algorithm.

For practicalities, the algorithm that achieved the best performances in the first experiment is used (i.e. Matching Networks). Matching Networks architecture is composed of a bidirectional Long-Short Term Memory (LSTM) [84] that is used to encode the support and the query sets as mentioned in the original paper [9]. The distance between all query images and normalized support images is on the other hand computed using the matrix of cosine similarities, and compute the query log probabilities based on the cosine similarity to support instances and support labels. Evaluation of the performances of the model also uses the confusion matrix with accuracy.

Optimization of Meta-Learning algorithms and evaluation of their ability to classify classes that were not seen during training

For this third experiment, the best-training method determined in the second experiment is used to fine-tune the best-performing Meta-Learning algorithm (i.e. Matching Networks) on pseudo-labeled data to evaluate its performances to correctly classify classes on the Darksound test set. Hyper-parameters of the model are optimized using the Python package *optuna*[9] for 100 trials to automatically optimize the model by comparing three different optimizers (Adam, RMSprop, SGD) methods with values of learning rate ranging from 0.00001 to 0.1. On the other hand, the Cross-Entropy loss is computed between the training data and the model's predictions as the cost function. To establish the performances of the model in a more thoroughly, a K-Fold Cross Validation is built on the Darksound dataset. This is done over 5 different folds in the shape of Few-Shot Image Classification tasks, where the average performance of the model is calculated on 100 episodic tasks in every fold, and the average score over all the folds corresponds to a more generic overview of the final performances of the model. Evaluation of the performances of the model also uses the confusion matrix with accuracy and additional performance metrics such as Precision, Recall, and F-1 score.

Performances metrics

Precision is used to evaluate the rate of correct predictions among the positive predictions. It is used to measure the capacity of the model not to make an error during a positive prediction and is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4.8}$$

[9] https://optuna.org/

Recall is used to evaluate the rate of all the positive ROIs detected by the model. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{4.9}$$

F-1 Score measures the ability of the model to predict positive ROIs, both in terms of Precision and Recall. It corresponds to the harmonic mean of these indicators and is calculated as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4.10}$$

Extraction of Meta-embeddings to cluster the data in the latent space and refine the clusters by fine-tuning Meta-Learning algorithms in an iterative way

For this last experiment, a comparison of the latent space representations (i.e. embeddings) extracted from the baseline model and the model fine-tuned on pseudo-labeled data are evaluated for their capacity to improve the quality of the clustering. Latent space representations correspond to vectors of 512 dimensions that are then normalized between 0 and 1 using the MinMaxScaler class from the pre-processing module of *scikit-learn*[10]. Finally, the normalized vectors are clustered using a density-based clustering algorithm.

Density-based algorithms are useful for merging areas with a high density of examples into clusters. These algorithms have the advantage of not having to specify the number of clusters and can handle noise while keeping it out of any cluster (e.g. DBSCAN, OPTICS). However, density-based algorithms generally have difficulty with high-dimensional and variable-density data. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [85] algorithm has recently allowed tackling this kind of problem by running the DBSCAN algorithm on different epsilon (ϵ) values. The values are evaluated in a hierarchical way which allows us to find a clustering with better stability on ϵ. In this thesis, the HDBSCAN Clustering library[11] is used to find clusters of different densities with the default parameters. For each iteration, outliers detected by the HDBSCAN algorithm are removed to construct Few-Shot Image Classification tasks using the labels

[10] https://scikit-learn.org/

[11] https://hdbscan.readthedocs.io/en/latest/index.html

of the predicted clusters. The number of clusters found by HDBSCAN represents the number of N-ways that are used for fine-tuning the model. The number of K-shot and Q-query samples is, on the other hand, determined by the minimum number of points allowed per cluster. Baseline performances of unsupervised embedding algorithms in the UML framework have shown that accuracy increases as the number of shots is increased (see subsection 3.2.1). Therefore, the *min_cluster_size* parameter of the HDBSCAN algorithm is set to 6 to allow the construction of Few-Shot Image Classification with a minimum of 5-shots, where $K = 5$ and $Q = 1$ because $K + Q$ need to be inferior or equal to the *min_cluster_size*. The model is fine-tuned for 20 epochs for each iteration to avoid over-fitting and clustering performance metrics are computed.

The MEC method is experimented with Meta-embeddings extracted from models that have been fine-tuned with different numbers of ways (i.e. 5-ways and 20-ways) to see if it can have an impact on the final number of clusters found. The MEC method is also experimented by artificially augmenting the number of samples found in the clusters by 50. For this purpose, initial waveform transformations introduced in subsection 4.1.1 are used except Pitch-Shift as it is assumed that there is not enough data in the clusters to find the averaged frequency of the acoustic units. The MEC method is finally evaluated by computing clustering performance metrics over the iterations and visualizing the quality of the clustering by reducing the dimensions of the feature vectors to two. For this purpose, the Uniform Manifold Approximation and Projection for Dimension Reduction[12] (UMAP) algorithm is used. An overview of the behaviors of the proposed framework is presented in Figure 4.9.

Clustering performance metrics

Clustering-based unsupervised methods can be evaluated using various performance metrics. In the field of Deep Clustering [86], three standard unsupervised evaluation metrics are generally used to indicate the average correct classification rate of clustering samples: Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). Regarding the validation of the quality of the clustering, the Density-Based Clustering Validation (DBCV) [87] metric is used for interpreting and validating the relative density connection between the clusters.

[12] https://umap-learn.readthedocs.io/en/latest/

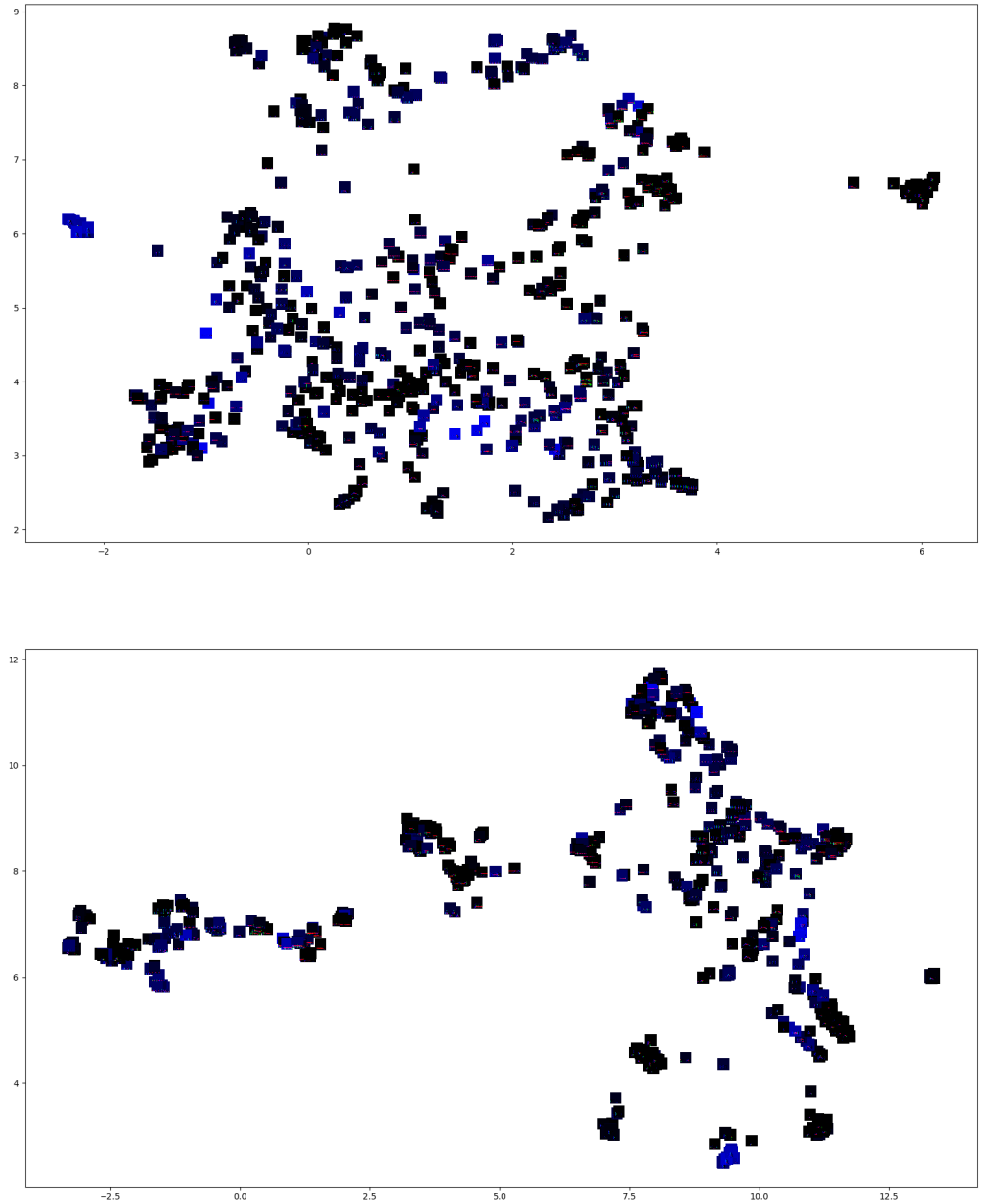


Figure 4.9: Visual comparison of the clustering quality in two dimensions using the UMAP algorithm. The clustering of the data is performed on the latent space with two different types of embedding using the HDBSCAN algorithm. Top: Baseline embeddings (ResNet18). Bottom: Meta-embeddings (Matching Networks) fine-tuned and optimized on the Darksound data set.

Accuracy for Clustering (ACC)

ACC differs from the standard classification accuracy metric because unsupervised clustering algorithms can potentially use a different cluster label than the actual ground truth label to represent the same cluster. Consequently, it is required to use a mapping m to represent the set of all possible permutations between the cluster labels and ground truth labels. In this experiment, ACC is used to find all possible one-to-one mappings m between the ground truth labels y and the cluster labels c, as defined in the equation 4.11.

$$\mathrm{ACC}(y, c) = max_m \frac{\sum_{i=1}^{n} 1\{y_i = m(c_i)\}}{n} \tag{4.11}$$

Normalized Mutual Information (NMI)

Recent works used the Normalized Mutual Information (NMI) metric for assessing the performance of a clustering model in the UML framework [48, 70]. In this experiment, NMI metric is computed after each iteration using the library *scikit-learn* to account for the entropy reduction of class labels based on the labels associated with the clusters. NMI score ranges from 0 to 1, where a 1 stands for the perfect alignment between two clusters. Note that the NMI score is independent of the permutation of labeling orders. The NMI score between clusters X and Y is defined in the equation 4.12, where I() corresponds to the Mutual Information metric and H() the entropy metric.

$$\mathrm{NMI}(X, Y) = \frac{\mathrm{I}(X, Y)}{\sqrt{\mathrm{H}(X)\mathrm{H}(Y)}} \tag{4.12}$$

Adjusted Rand Index (ARI)

ARI metric is also computed after each iteration using the library *scikit-learn* to measure the similarity between clusters, although ignoring permutations. Compared to the Rand Index (RI) score, the ARI score is adjusted for the number of samples and the number of clusters. It is defined in equation 4.13.

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}(\mathrm{RI})}{\max(\mathrm{RI}) - \mathbb{E}(\mathrm{RI})} \tag{4.13}$$

As for the NMI score, a perfect ARI labeling is scored 1.0. However, the ARI score ranges from -1 to 1, where a value of 0 indicates random agreement and a value of -1 shows complete disagreement.

Density-Based Clustering Validation (DBCV)

To evaluate clustering in an unsupervised way (i.e. without using ground truth labels), it is common to use objective metrics to measure the distance or cohesion between the clusters. However, most of the metrics or indexes used for this kind of task (e.g. Silhouette score) do not take noise into account and are therefore inappropriate for measuring density-based clustering techniques (i.e. HDBSCAN). To tackle this problem, the quality of clustering is validated using a Python implementation of DBCV[13] as this metric has the advantage of taking noise into account and capturing the shape property of clusters through densities and not distances. The DBCV metric corresponds to the weighted average of the values of the "Validity Index" of all the clusters in C, where $C = \{C_i\}, 1 \leq i \leq l$. The DBCV is defined in equation 4.14.

$$\text{DBCV}(C) = \sum_{i=1}^{i=l} \frac{|C_i|}{|O|} V_C(C_i) \tag{4.14}$$

DBCV score ranges from -1 to 1, the closer the value is to 1, the better the quality of the density-based clustering.

[13] https://github.com/christopherjenness/DBCV

Chapter 5

Results & Discussion

In this chapter, the results including the comparison of performances of Meta-Learning algorithms and training methods are first introduced. Moreover, the results related to the classification performances of optimized Meta-Learning algorithms on the Darksound dataset, as well as the clustering performances of the MEC method are presented. In the second step, the challenges related to the Few-Shot Image Classification and the clustering of rare tropical bird species are discussed, notably the environmental considerations regarding the computational resources needed in ecoacoustics with DL and some future work propositions related to our research.

5.1 Results

5.1.1 Meta-Learning Algorithms

Table 5.1 presents a comparison of the raw performances of the three Meta-Learning algorithms used in this thesis, namely: Matching Networks [9], Prototypical Networks [10], and Relation Networks [59]. Baseline models were compared to models that were fine-tuned on the Darksound training set using episodic training. Results show that fine-tuning a model on a pseudo-labeled dataset allows, in all cases, to improve the performance of the model. This has the advantage to tackle a recurring problem in ecoacoustic projects that is associated with the lack of large labeled datasets. On the other hand, Matching Networks was identified as the best-performing model because it achieved the best results in almost every Few-Shot Image Classification task.

	5-ways Acc.		**20-ways Acc.**	
Algorithm	1-shot	5-shots	1-shot	5-shots
Matching (*baseline*)	68.56%	84.30%	45.74%	65.46%
Matching (*fine-tune*)	74.88%	**87.48%**	**59.13%**	**78.98%**
Prototypical (*baseline*)	66.12%	84.90%	42.20%	64.79%
Prototypical (*fine-tune*)	**78.80%**	86.79%	58.54%	76.99%
Relation (*baseline*)	-	-	-	-
Relation (*fine-tune*)	71.94%	84.53%	50.09%	70.63%

Table 5.1: Results for 5 and 20 ways Few-Shot Classification tasks on the Darksound dataset. All the *baseline* models correspond to a pre-trained ResNet18 with Fully Connected (FC) layers replaced with a Meta-Learning algorithm. All the *fine-tuned* models correspond to the fine-tuning of *baseline* models for 20 epochs, with 1 epoch corresponding to 100 episodic tasks on a random Few-Shot batch. A scheduler is used for reducing the learning rate by a factor of 0.1 when an indicator has stopped improving after 5 epochs. An early stop is applied if the performance of the model has not improved after 10 epochs. Each evaluation result corresponds to the average accuracy of a 5-Fold Cross Validation on 100 episodic tasks.

5.1.2 Episodic/Classical Training

Table 5.2 presents a comparison of the performances of episodic versus classical training for models that were fine-tuned for various Few-Shot Image Classification tasks. As can be seen from the results, using episodic training against classical training allows the model to perform better in almost all the Few-Shot Classification tasks, although classical training performs better on the 5-ways-5-shots task. Consequently, episodic training was preferred over classical training for our third experiment because it improved the model performance and potentially contributed to producing representative Meta-embeddings for clustering bird songs in soundscape recordings.

5.1.3 Few-Shot Image Classification

Table 5.3 presents the average accuracy, precision, recall, and F-1 scores of the models that were fine-tuned and optimized for 20-way Few-Shot Image Classification tasks on the Darksound dataset. All the results correspond to the average performances of the models in a 5-Fold Cross-Validation procedure, where one fold represents the average performance on 100 episodic tasks.

	5-ways Acc.		20-ways Acc.	
Training	1-shot	5-shots	1-shot	5-shots
Episodic	**74.88%**	87.48%	**59.13%**	**78.98%**
Classical	71.28%	**91.86%**	53.08%	78.33%

Table 5.2: Comparison of episodic versus classical training with Meta-Learning algorithm (Matching Networks [9]) fine-tuned on the Darksound dataset. The training methods are evaluated for 20 epochs with the Adam optimizer (learning rate = 0.0001). A scheduler is applied to reduce the learning rate when the training loss stopped improving after 5 epochs, as well as an early stop if the validation accuracy of the model has not improved after 10 epochs. In every case, the batch size used for training the model corresponds to the Few-Shot Classification task (e.g. 5-ways-5-shots-1-query = 5 $\times$ (5 + 1) = 30). Each evaluation result corresponds to the average accuracy of a 5-Fold Cross Validation on 100 episodic tasks.

20-ways-1-shot				
Algorithm	Accuracy	Precision	Recall	F1-Score
Baseline	45.74%	46.77%	45.74%	45.63%
Fine-tune	*59.13%*	*60.97%*	*60.03%*	*59.99%*
Optimized	**62.16%**	**62.90%**	**62.17%**	**61.94%**
20-ways-5-shots				
Algorithm	Accuracy	Precision	Recall	F1-Score
Baseline	65.46%	66.16%	65.46%	64.67%
Fine-tune	*78.98%*	*79.30%*	*78.98%*	*78.91%*
Optimized	**79.90%**	**80.33%**	**79.90%**	**79.81%**

Table 5.3: Classification performances of the Matching Networks fine-tuned and optimized for 20-ways Few-Shot Classification tasks. All the results correspond to the average performances of the models on the Darksound dataset in a 5-Fold Cross-Validation procedure, where one fold represents the average performance on 100 episodic tasks. The models were fine-tuned using the optimizer Adam and a Cross-Entropy loss and optimized using the Python library *optuna* for 100 trials to automatically optimize the model by comparing three different optimizers methods (Adam, RMSprop, SGD) with values of learning rate ranging from 0.00001 to 0.1.

For the 20-ways-1-shot classification task, the best trial using *optuna* set the learning rate of the optimizer Adam to 0.000024 and obtained an

average accuracy of 62.16% on the Darksound test set. This represents an improvement of ≈ 16.42% or ≈ 3.03% compared to the baseline model or fine-tune model respectively. For the 20-ways-5-shots classification task, the best trail using *optuna* set the optimizer Adam with a learning rate of 0.000028 and allowed to obtain an average accuracy of 79.90% on the Darksound data set. That is ≈ 14.44% or ≈ 0.92% improvement compared to the baseline model or the fine-tuned model respectively. All in all, the Adam optimizer gave the best results with a slightly different learning rate ≈ 0.000024 − 28 compared to the default 0.0001 learning rate value. Results show that fine-tuning and optimizing a model on pseudo-labeled data in an episodic way can significantly improve the classification of bird songs in soundscape recordings and tackle the problem of the Few-Shot Image Classification. Training and validation curves for the optimization of the models are presented in Figure 5.1.

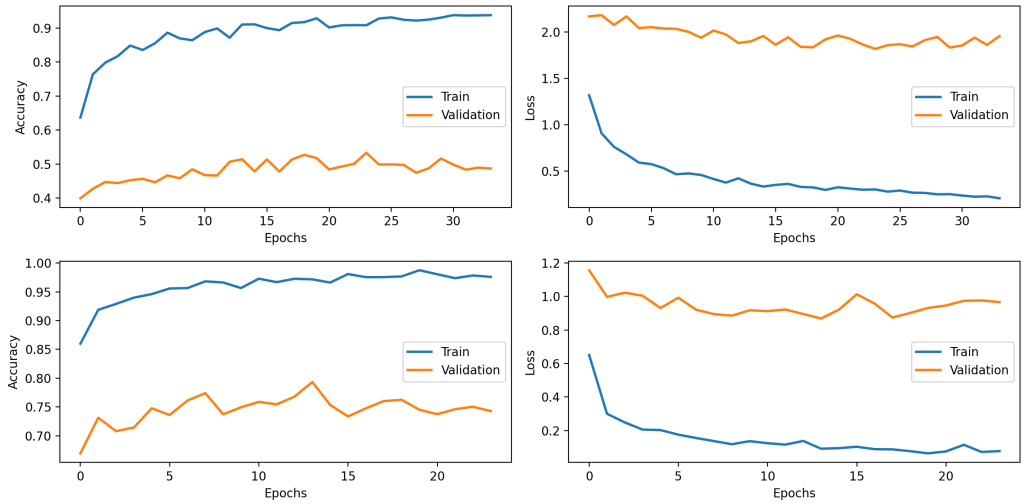


Figure 5.1: Training and validation curves for the optimization of models trained using the Darksound dataset. Top: Accuracy and loss for model training and validation on a 20-ways-1-shot task. Bottom: Accuracy and loss for model training and validation on a 20-ways-5-shots task.

5.1.4 Meta Embedded Clustering

Table 5.4 presents a comparison of the clustering performances of the baseline embeddings versus Meta-embeddings. Meta-embeddings were extracted from models fine-tuned on pseudo-labeled data and optimized for a 20-ways-5-shots classification task. A visual representation of the number of clusters

found by the HDBSCAN algorithm in the latent space is illustrated in the Figure 5.2. Results show that using Meta-embeddings fine-tuned on pseudo-labeled data can significantly improve the accuracy of the clustering (30.58% vs. 67.48%) as well as allow to get closer to the actual number of clusters to be determined (4 vs. 13 for 21 bird species to be found).

Embedding	Number of clusters	Accuracy	NMI	ACI	DBCV
Baseline	4	30.58%	0.1201	0.0212	-0.0949
Meta	**13**	**67.48%**	**0.8142**	**0.5813**	**-0.2029**

Table 5.4: Comparison of the baseline embeddings versus Meta-embeddings fine-tuned on pseudo-labeled data and optimized for a 20-ways-5-shots classification task. Data clustering is performed on the latent space using the HDBSCAN algorithm with *min_cluster_size* parameter set to 6. Results are obtained by evaluating the ground truth labels to the labels predicted by HDBSCAN using clustering performance metrics (Accuracy, NMI, ACI, and DBCV).

Initial and final results found with the MEC method over 20 iterations are presented in Table 5.5. Initial results of the MEC method (i.e. iteration 0) correspond to the initial clustering performances performed on the latent space given by the best Meta-Learning model (i.e. Matching Networks) fine-tuned and optimized on pseudo-labeled data, but without learning latent space representations from the clusters predicted by the HDBSCAN algorithm. After each iteration, the model is fine-tuned again with the pseudo-labeled data assigned by the last clustering process. Data labeled as noise are removed during the refining of the model. The final results of the MEC method are calculated by taking the average number of clusters found over the 20 iterations, where the highest DBCV score associated to the average number of clusters determines the final accuracy of the MEC method found at iteration n.

For this experiment, the objective was also to determine if using models that have been fine-tuned with a different number of ways (i.e. 5-ways and 20-ways) could have an impact on the clustering performances. Results show that when using Meta-embeddings with models fine-tuned on different Few-Shot Image Classification tasks, different numbers of clusters were found. Nevertheless, the difference in the number of clusters found between the two experiments is relatively small (17 vs. 19). Even though the goal was to find a target number of 21 species, the results show that the model can improve its ability to determine the final number of clusters through-

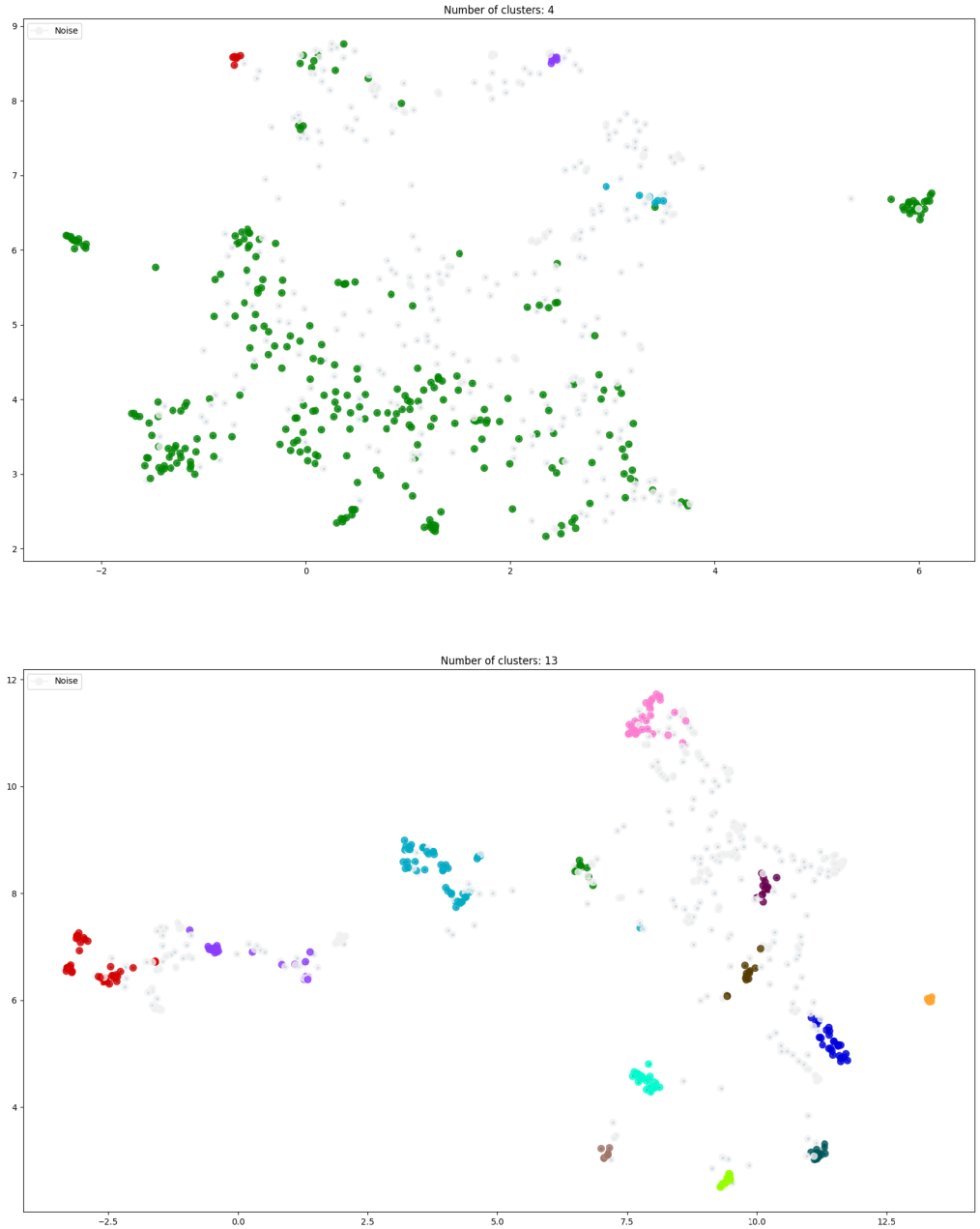


Figure 5.2: Visual representation of the number of clusters found by the HDBSCAN algorithm in the latent space. Data clustering is performed on two different types of embedding. Top: Baseline embeddings (ResNet18). Bottom: Meta-embeddings (Matching Networks) fine-tuned on pseudo-labeled data and optimized for a 20-ways-5-shots classification task.

5-ways-5-shots					
Iteration	Number of clusters	Accuracy	NMI	ACI	DBCV
0	17	69.10%	0.8460	0.5650	-0.3547
16	**19**	**76.60%**	**0.8681**	**0.6842**	**-0.0920**
20-ways-5-shots					
Iteration	Number of clusters	Accuracy	NMI	ACI	DBCV
0	13	67.48%	0.8142	0.5813	-0.2029
12	17	70.96%	0.8564	0.6153	-0.3547

Table 5.5: Results of the Meta Embedded Clustering (MEC) method using Meta-embeddings pre-trained on two Few-Shot Classification tasks. Clustering performance metrics are computed at each iteration over a total number of 20 iterations. For each iteration, Meta-embeddings are fine-tuned for 20 epochs on the predicted labels found by the HDBSCAN algorithm (i.e. without outliers). Results at iteration 0 indicate the initial clustering performance without learning the latent space representations from the predicted clusters. Results at iteration n are determined according to the highest DBCV score found for the average number of clusters determined over all iterations.

out iterations, as in both cases, the initial number of clusters found versus the final number of clusters found is closer to the actual number of clusters. Interestingly, using Meta-embeddings extracted from models that had been fine-tuned on 5-ways-5-shots tasks compared to Meta-embeddings extracted from models that had been fine-tuned on 20-ways-5-shots tasks allowed to obtain better performances for the MEC method (76.60% vs. 70.96%). Regarding the experiment of the MEC method with data augmentation, Figure 5.3 presents the behaviors of iterative clustering with and without data augmentation using Meta-embeddings extracted from models that had been fine-tuned for 5-ways-5-shots classification tasks. Compared to performing the MEC method without data augmentation, results show that augmenting the pseudo-labeled data did not improve the clustering accuracy nor the clustering quality, although it did smooth out the curves, especially in terms of determining the final number of clusters (Figure 5.3 (5)).

5.2 Discussion

This section discusses the proposed framework that is related to Few-Shot Image Classification and bird song clustering in soundscape recordings. Environmental considerations are also highlighted concerning the impact of de-

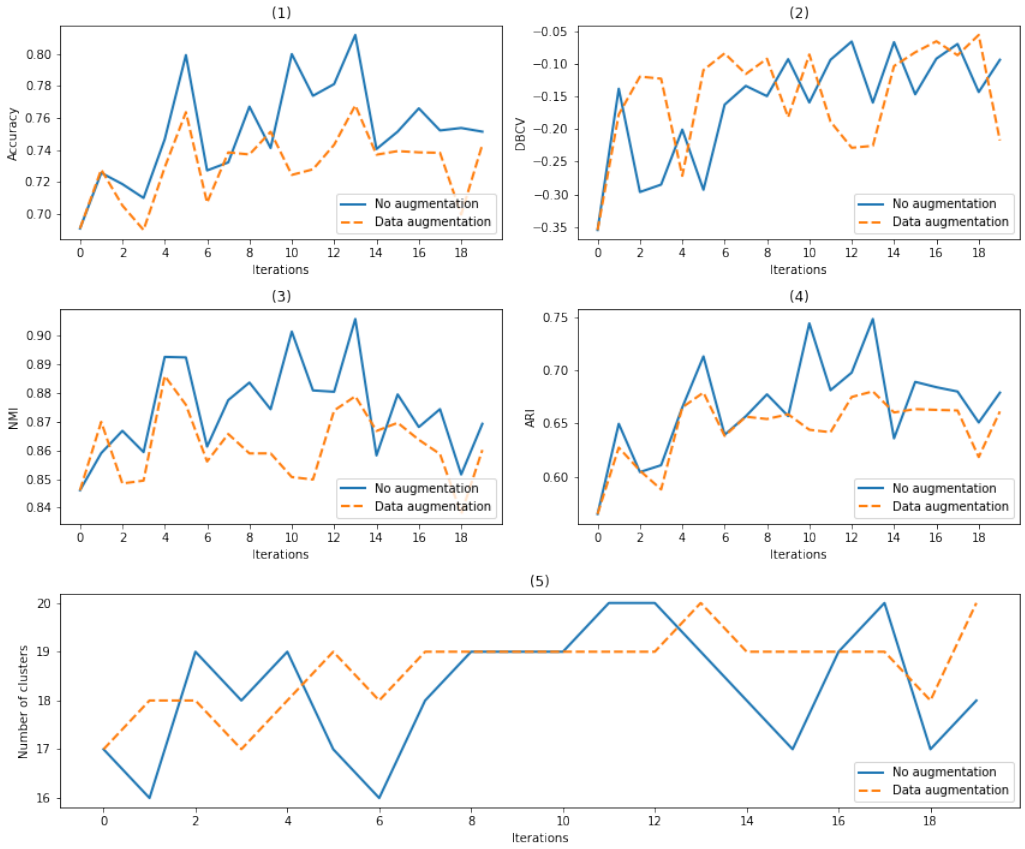


Figure 5.3: Behaviors of Meta Embedded Clustering (MEC) over 20 iterations. For each iteration, clustering performances are computed to evaluate the clustering quality of Meta-embeddings extracted from models fine-tuned on 5-ways-5-shots episodic tasks. (1) Accuracy (ACC) for clustering representing the best mapping between the ground truth labels and the cluster labels (2) Density-Based Clustering Validation (DBCV) interpreting the improvement of clustering quality over the iterations (3) Normalized Mutual Information (NMI) representing the entropy reduction of class labels based on the labels associated with the clusters. (4) Adjusted Rand Index (ARI) measuring the similarity between clusters. (5) Number of clusters found over the 20 iterations.

ploying such technology when it comes to their energy consumption. Furthermore, proposals for future work in connection with an ecoacoustic mission in real conditions are presented.

5.2.1 Meta-Learning Baseline

In this thesis, it was first necessary to define a baseline Meta-Learning algorithm to facilitate the evaluation of the proposed framework. The selection of the algorithm used was based on the accuracy and the ability of the algorithm to generalize to the Darksound test set. The results obtained in table 5.1 were used to define the Matching Networks as the baseline algorithm. At this stage, the algorithms were tested without considering parameter tuning, nevertheless, models could have produced better results if all parameters had been carefully tuned. However, the problem is that this would have required a different setting for each of the Few-Shot Classification tasks. This was not considered since Meta-Learning algorithms are primarily designed to be flexible in configurations that are not highly dependent on tuning.

5.2.2 Proposed Framework

To assess the extent to which the proposed framework addresses our research questions, we will now return to each of the initial questions and discuss them in detail.

Q1: How well does episodic training improve the performance of a Meta-Learning algorithm compared to classical training?

The results presented in Table 5.2 allowed us to answer our first research question and favor episodic learning over classical learning. However, the results show that this trend seems to be contradicted as the number of shots increases, as classical learning performs better for the 5-ways-5-shots classification task and almost as good for the 20-ways-5-shots classification task. Indeed, recent work suggests that competitive results can be obtained from classical training with simple Cross-Entropy loss compared to episodic training [49, 50]. Nevertheless, it is important to mention that our results evaluate the performance of two training methods on common grounds, which is not always the case in the literature. This implies using the same batch size for the training in both situations, given that this can have a critical influence in DL [83]. Indeed, unlike episodic training, classical training allows to use batch sizes that are independent of the number of classes, therefore, the optimization of this parameter in classical training can surely improve the performance of the model. In our case, the performances of these two training methods were compared on common bases, since it was assumed that this parameter could have an important influence on the results.

Q2: To what extent can Meta-Learning algorithms fine-tuned on pseudo-labeled data classify classes that were not used during training?

The results presented in Table 5.3 allowed us to conclude that fine-tuning a Meta-Learning algorithm on pseudo-labeled data can largely improve the performance of the model. In the context of ecoacoustics, and more specifically in the detection and classification of bird species, this makes it possible to create efficient classification models without the need to annotate a large dataset. Although here classification is performed in a *closed-set* setting, the democratization of this kind of practice in *open-set* settings could greatly facilitate the classification of bird species whose vocalizations are well represented in the Xeno-Canto database. On the other hand, the diagnosis of the model performance with the help of the learning curves presented in Figure 5.1 has also allowed us to highlight a problem related to the Darksound training set. Indeed, this dataset does not seem to provide enough information to learn the problem given the important gap that remains between the training and validation curves. This may be related to the fact that the training set contains features with lower variance than the validation set. Thus, adding more samples or increasing the number of augmented samples could help tackle the problem related to the variability of the features in the training data.

Q3: To what extent Meta-embeddings can improve the clustering quality of unlabeled data?

When performing clustering on the latent space, results presented in Table 5.4 have shown that Meta-embeddings compared to baseline embeddings improve the quality of the clustering. In addition, results presented in Table 5.5 have shown that performing iterative clustering on the Meta-embeddings can also help refine the clusters and further improve the clustering quality. However, how Meta-embeddings were fine-tuned beforehand has also shown that it can influence the results. Indeed, using 5-ways-5-shots Meta-embeddings compared to 20-ways-5-shots Meta-embeddings allowed us to obtain better clustering performances. It thus appears more appropriate to extract Meta-embeddings from a model that obtained better classification results even if the number of ways used for the fine-tuning is closer to the actual number of clusters to determine. On the other hand, although augmenting the data usually helps to prevent over-fitting and improve the accuracy of the model, this has not been confirmed when performing the MEC method. The reason is perhaps that the initial clusters found by the HDBSCAN algorithm in the

latent space are not accurate enough, consequently, fine-tuning the model on pseudo-labeled data that is misclassified can lead the model in the wrong direction even further if it appears that the data has been augmented. A way to get around this problem could be to augment the data after a few iterations to give the model time to become more confident in its choices.

5.2.3 Environmental considerations

The deployment of large-scale computing in the context of ecoacoustics with DL often requires the use of many technologies with a significant environmental impact. We thus must take this impact into account and to evaluate its energy consumption as well as the use of resources on which it depends (e.g. rare earth minerals or electronic waste considerations). A summary of these considerations has been introduced in [88] and proposes, among others, the use of battery-less ecoacoustic devices. In this thesis, an estimate of the energy consumption related to the training and validation of the models for Few-Shot Image Classification tasks has been calculated using the Python package *PyJoules*[1]. This made it possible to define the total energy consumed for the fine-tuning of the optimized models on a machine with 28 cores (Intel Xeon)/128 and a GPU NVIDIA GeForce GTX 1080 Ti. By default, the values obtained with the package *PyJoules* are expressed in microjoules (μJ). These were converted into Watts (W) by recovering the training time (t) in seconds recorded for each epoch. Thus, the equation to define the energy consumption in Watts per epoch (We) was defined as follows:

$$We = \frac{(\mu J \times 1^{-6})}{t} \tag{5.1}$$

An overview of the energy consumption for 20 epochs is shown in Figure 5.4. This allows us to point out a higher energy consumption when a larger amount of data is involved in the training phase (1 shot vs. 5 shots). Thus, we believe that the systematic use of units of power such as We should also be taken into account in the future optimization of the models, since to be really *efficient*, a model must have good classification performances as well as low energy consumption, especially when the purpose of the analysis is related to environmental preservation and biodiversity conservation.

5.2.4 Future Work

To continue developing the proposed framework in the future, it is planned to test it on data collected during an ecoacoustic mission in real-conditions.

[1] https://pypi.org/project/pyJoules/

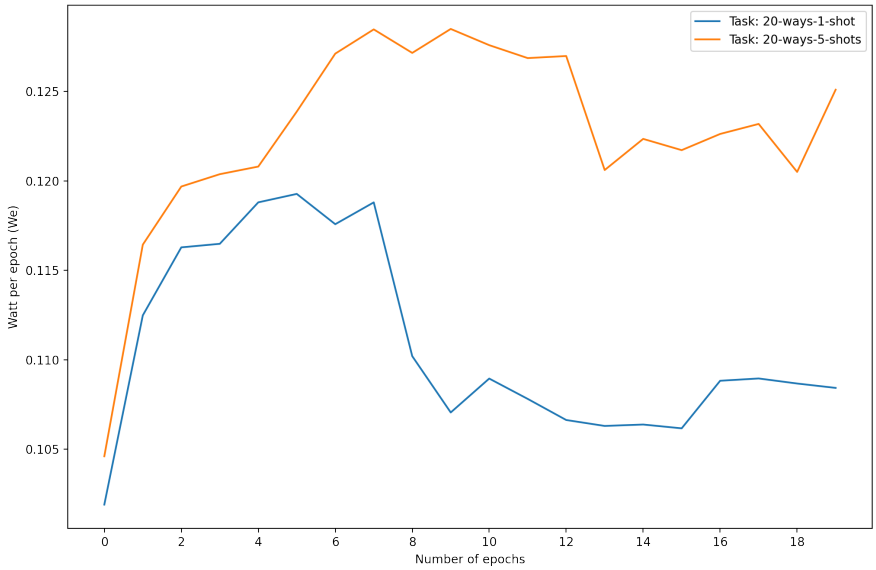


Figure 5.4: Energy consumption in Watt per epochs (We) for 20-ways Few-Shot Classification tasks. Energy consumption is estimated using the Python package *PyJoules* for the fine-tuning of the models on a machine with 28 cores (Intel Xeon)/128 and a GPU NVIDIA GeForce GTX 1080 Ti.

To this end, the EAR team of the MNHN of Paris collected a dataset in French Guyana between mid-December and mid-May 2022. This dataset is composed of 6800 hours of audio recordings to map the presence/absence of nocturnal and crepuscular species in the area. The recordings focus on time periods where the target species are most likely to vocalize (high probability of capturing a vocalization). This corresponds to 1h of recording before dawn (i.e. civil twilight), 30min centered on sunset and 30 min centered on the beginning of the night (about 1h15 after sunset) to focus on the truly nocturnal species, i.e. 2h/d. Nevertheless, to date, no annotations of the data have been performed due to the large amount of data and the need to find experts able to identify the target species. Therefore, our future objective is to test our proposed framework on this dataset to improve the visualization of the inherent structure of the data that will be more easily identified by experts as clusters, without the need to label each record one by one. This would save valuable time and, ideally, allow for improved quality of data clustering for the labeling of vocalizations of target or unknown species.

Chapter 6

Conclusion

A recurrent problem in ecoacoustic projects is the lack of large labeled datasets available for training models. The primary objective of this master thesis has accordingly been to discover efficient ways to respond to this problem using learning methods with good generalization capacities and that can adapt easily to new learning tasks. The use of the Meta-Learning framework has consequently proven to be convenient in dealing with the problem of Few-Shot Image Classification. In this thesis, several Meta-Learning algorithms based on metric-learning strategies have been tested to define a reference model for our further experiments.

The global objective of this research has been to facilitate the work of ecoacousticians in their management of acoustic data and identification of potential new taxa, by discovering and gradually improving the inherent structure of unlabeled data. The numerous tests carried out in this thesis have shown that Matching Networks are the most suitable Meta-Learning algorithm for the proposed framework. Moreover, the use of the episodic learning method compared to the classical learning method has proven to allow an improvement of the models performances. Taking into consideration the total number of solutions considered in this thesis, it was the aforementioned combination that produced the highest total accuracy for the majority of the configurations, including an accuracy of 79.81% for the classification of 20 tropical nocturnal and crepuscular bird species from the Darksound dataset in a 5-shot classification task. In the second step, this learning framework permitted an extraction of more meaningful latent space representations when clustering similar unlabeled data. As a matter of fact, the fine-tuning of the models on pseudo-labeled data allowed us to improve the performance of the data clustering by 36.90%, compared to the simple use of latent space representations extracted from models commonly used in com-

puter vision (ResNet18). Based on the unsupervised clustering-based methods reviewed in our theoretical background, the Meta Embedded Clustering (MEC) method turned out to progressively improve the inherent structure of unlabeled data. This method has eventually allowed us to further improve the accuracy of the data clustering (69.10% vs. 76.60%) and, in this way, contribute to determine a number of clusters closer to the actual number of clusters expected.

In conclusion, the use of unsupervised Meta-embedding has proven to be an effective solution for improving the clustering of bird songs in soundscape recordings. These technological methods can therefore bring forward novel research in developing countries that can facilitate the identification of species as well as the detection of potential new rare bird species.

References

[1] Judith C Brown and Paris Smaragdis. "Hidden Markov and Gaussian mixture models for automatic call classification". In: *The Journal of the Acoustical Society of America* 125.6 (2009), EL221–EL224.

[2] Len Thomas and Tiago A Marques. "Passive acoustic monitoring for estimating animal density". In: *Acoustics Today* 8.3 (2012), pp. 35–44.

[3] Todor Ganchev. *Computational bioacoustics: Biodiversity monitoring and assessment*. Vol. 4. Walter de Gruyter GmbH & Co KG, 2017.

[4] Jérôme Sueur and Almo Farina. *Ecoacoustics: the ecological investigation and interpretation of environmental sound. Biosemiotics 8, 493–502*. 2015.

[5] Dan Stowell. "Computational bioacoustic scene analysis". In: *Computational analysis of sound scenes and events*. Springer, 2018, pp. 303–333.

[6] Irina Tolkova et al. "Parsing birdsong with deep audio embeddings". In: *arXiv preprint arXiv:2108.09203* (2021).

[7] Vincent Lostanlen et al. "Birdvox-full-night: A dataset and benchmark for avian flight call detection". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 266–270.

[8] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

[9] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016).

[10] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[11] Adam Santoro et al. “Meta-learning with memory-augmented neural networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 1842–1850.

[12] Dan Stowell et al. “Detection and classification of acoustic scenes and events”. In: *IEEE Transactions on Multimedia* 17.10 (2015), pp. 1733–1746.

[13] Theodore A Parker III. “On the use of tape recorders in avifaunal surveys”. In: *The Auk* 108.2 (1991), pp. 443–444.

[14] Jérôme Sueur et al. “Rapid acoustic survey for biodiversity appraisal”. In: *PloS one* 3.12 (2008), e4065.

[15] Dan Stowell et al. “Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge”. In: *Methods in Ecology and Evolution* 10.3 (2019), pp. 368–380.

[16] Vincent Lostanlen et al. “Per-channel energy normalization: Why and how”. In: *IEEE Signal Processing Letters* 26.1 (2018), pp. 39–43.

[17] Paola Laiolo. “The emerging significance of bioacoustics in animal species conservation”. In: *Biological conservation* 143.7 (2010), pp. 1635–1645.

[18] Juan Sebastian Ulloa et al. “Estimating animal acoustic diversity in tropical environments using unsupervised multiresolution analysis”. In: *Ecological Indicators* 90 (2018), pp. 346–355.

[19] Daniel T Blumstein et al. “Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus”. In: *Journal of Applied Ecology* 48.3 (2011), pp. 758–767.

[20] Mallory M Morgan and Jonas Braasch. “Open set classification strategies for long-term environmental field recordings for bird species recognition”. In: *The Journal of the Acoustical Society of America* 151.6 (2022), pp. 4028–4038.

[21] Tiago Fernandes Tavares. “Open-set classification approaches to automatic bird song identification: towards non-invasive wildlife monitoring in Brazilian fauna”. In: *IEEE Latin America Transactions* 20.11 (2022), pp. 2388–2394.

[22] Xavier Anguera, Chuck Wooters, and Javier Hernando. “Acoustic beamforming for speaker diarization of meetings”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.7 (2007), pp. 2011–2022.

[23] Dan Stowell. “Computational bioacoustics with deep learning: a review and roadmap”. In: *PeerJ* 10 (2022), e13152.

[24] David Chesmore. “Automated bioacoustic identification of species”. In: *Anais da Academia Brasileira de Ciências* 76 (2004), pp. 436–440.

[25] Miguel A Acevedo and LUIS J VILLANUEVA-RIVERA. “From the field: Using automated digital recording systems as effective tools for the monitoring of birds and amphibians”. In: *Wildlife Society Bulletin* 34.1 (2006), pp. 211–214.

[26] Marc O Lammers et al. “An ecological acoustic recorder (EAR) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats”. In: *The Journal of the Acoustical Society of America* 123.3 (2008), pp. 1720–1728.

[27] Alexis Joly et al. “Overview of lifeclef 2022: an evaluation of machine-learning based species identification and species distribution prediction”. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2022, pp. 257–285.

[28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[29] Tong Chen et al. “Deepcoder: A deep neural network based video compression”. In: *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE. 2017, pp. 1–4.

[30] Liwen You et al. “Transformer-based bioacoustic sound event detection on few-shot learning tasks”. In: *Amazon Science* (2023).

[31] Juan Colonna et al. “Automatic classification of anuran sounds using convolutional neural networks”. In: *Proceedings of the ninth international c* conference on computer science & software engineering*. 2016, pp. 73–78.

[32] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[33] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[34] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[35] Sven Heuer et al. *New aspects in birdsong recognition utilizing the gabor transform*. Universitätsbibliothek der RWTH Aachen Aachen, 2019.

[36] Ivan Himawan et al. "Deep Learning Techniques for Koala Activity Detection." In: *INTERSPEECH*. 2018, pp. 2107–2111.

[37] Jie Xie et al. "Investigation of different CNN-based models for improved bird sound classification". In: *IEEE Access* 7 (2019), pp. 175353–175361.

[38] Vincent Lostanlen et al. "Robust sound event detection in bioacoustic sensor networks". In: *PloS one* 14.10 (2019), e0214168.

[39] Lonce Wyse. "Audio spectrogram representations for processing with convolutional neural networks". In: *arXiv preprint arXiv:1706.09559* (2017).

[40] Mario Lasseck. "Audio-based Bird Species Identification with Deep Convolutional Neural Networks." In: *CLEF (working notes)* 2125 (2018).

[41] Honglie Chen et al. "Vggsound: A large-scale audio-visual dataset". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 721–725.

[42] Richard Bellman. "On the reduction of dimensionality for classes of dynamic programming processes". In: *Journal of Mathematical Analysis and Applications* 3.2 (1961), pp. 358–360.

[43] Veronica Morfi and Dan Stowell. "Deep learning for audio event detection and tagging on low-resource datasets". In: *Applied Sciences* 8.8 (2018), p. 1397.

[44] Jack LeBien et al. "A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network". In: *Ecological Informatics* 59 (2020), p. 101113.

[45] Alexei Baevski et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems* 33 (2020), pp. 12449–12460.

[46] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[47] Archit Parnami and Minwoo Lee. "Learning from few examples: A summary of approaches to few-shot learning". In: *arXiv preprint arXiv:2203.04291* (2022).

[48] Zilong Ji et al. "Unsupervised few-shot feature learning via self-supervised training". In: *Frontiers in computational neuroscience* 14 (2020), p. 83.

[49] Wei-Yu Chen et al. “A closer look at few-shot classification”. In: *arXiv preprint arXiv:1904.04232* (2019).

[50] Carlos Medina, Arnout Devos, and Matthias Grossglauser. “Self-supervised prototypical transfer learning for few-shot classification”. In: *arXiv preprint arXiv:2006.11325* (2020).

[51] Yu Wang et al. “Few-shot drum transcription in polyphonic music”. In: *arXiv preprint arXiv:2008.02791* (2020).

[52] Yu Shiu et al. “Deep neural networks for automated detection of marine mammal species”. In: *Scientific reports* 10.1 (2020), p. 607.

[53] I Nolasco et al. “Few-shot bioacoustic event detection at the DCASE 2022 challenge”. In: *arXiv preprint arXiv:2207.07911* (2022).

[54] Jigang Tang et al. “Few-shot embedding learning and event filtering for bioacoustic event detection”. In: *iFLYTEK Research Institute, Hefei, China, Tech. Rep* (2022).

[55] Haohe Liu et al. “Surrey system for dcase 2022 task 5: Few-shot bioacoustic event detection with segment-level metric learning”. In: *arXiv preprint arXiv:2207.10547* (2022).

[56] John Martinsson et al. “Few-shot bioacoustic event detection using a prototypical network ensemble with adaptive embedding functions”. In: *Detection and Classification of Acoustic Scenes and Events 2022, DCASE 2022*. 2022.

[57] Michael Hertkorn and ZF Friedrichshafen AG. “Few-shot bioacoustic event detection: Don’t waste information”. In: *ZF Friedrichshafen AG, Friedrichshafen, Germany, Tech. Rep* (2022).

[58] Miao Liu et al. “Bit srcb team’s submission for dcase2022 task5-few-shot bioacoustic event detection”. In: *Beijing Institute of Technology, Beijing, China, Tech. Rep* (2022).

[59] Flood Sung et al. “Learning to compare: Relation network for few-shot learning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.

[60] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.

[61] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*. Springer. 2015, pp. 84–92.

[62] Michelangelo Acconcjaioco and Stavros Ntalampiras. "One-shot learning for acoustic identification of bird species in non-stationary environments". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 755–762.

[63] Kyle Hsu, Sergey Levine, and Chelsea Finn. "Unsupervised learning via meta-learning". In: *arXiv preprint arXiv:1810.02334* (2018).

[64] Shuo Li et al. "Unsupervised Few-Shot Image Classification by Learning Features into Clustering Space". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*. Springer. 2022, pp. 420–436.

[65] Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. "Unsupervised meta-learning for few-shot image classification". In: *Advances in neural information processing systems* 32 (2019).

[66] Xingping Dong, Jianbing Shen, and Ling Shao. "Rethinking Clustering-Based Pseudo-Labeling for Unsupervised Meta-Learning". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*. Springer. 2022, pp. 169–186.

[67] Xi Chen et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in neural information processing systems* 29 (2016).

[68] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning". In: *arXiv preprint arXiv:1605.09782* (2016).

[69] David Berthelot et al. "Understanding and improving interpolation in autoencoders via an adversarial regularizer". In: *arXiv preprint arXiv:1807.07543* (2018).

[70] Mathilde Caron et al. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 132–149.

[71] Félix Michaud et al. "Unsupervised classification to improve the quality of a bird song recording dataset". In: *Ecological Informatics* 74 (2023), p. 101952.

[72] Juan Sebastián Ulloa et al. "scikit-maad: an open-source and modular toolbox for quantitative soundscape analysis in Python". In: *Methods in Ecology and Evolution* 12.12 (2021), pp. 2334–2340.

[73] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 1996, pp. 226–231.

[74] Derry Fitzgerald. “Harmonic/percussive separation using median filtering”. In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. Vol. 13. 2010, pp. 1–4.

[75] Jonathan Driedger, Meinard Müller, and Sascha Disch. “Extending Harmonic-Percussive Separation of Audio Signals.” In: *ISMIR*. 2014, pp. 611–616.

[76] Thailsson Clementino and Juan Colonna. “Using Triplet Loss for Bird Species Recognition on BirdCLEF 2020.” In: *CLEF (Working Notes)*. 2020.

[77] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[78] Mario Lasseck. “Bird Species Identification in Soundscapes.” In: *CLEF (Working Notes)* 2380 (2019).

[79] Juliette Florentin, Thierry Dutoit, and Olivier Verlinden. “Detection and identification of European woodpeckers with deep convolutional neural networks”. In: *Ecological Informatics* 55 (2020), p. 101023.

[80] Xiaoxu Li et al. “Deep Metric Learning for Few-Shot Image Classification: A Review of Recent Developments”. In: *Pattern Recognition* (2023), p. 109381.

[81] S. Prince W. Zi L. S. Ghoraie. *Few-shot learning and meta-learning*. https://www.borealisai.com/research-blogs/tutorial-2-few-shot-learning-and-meta-learning-i/. Accessed: 2023-02-18.

[82] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[83] Nitish Shirish Keskar et al. “On large-batch training for deep learning: Generalization gap and sharp minima”. In: *arXiv preprint arXiv:1609.04836* (2016).

[84] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[85] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. "Density-based clustering based on hierarchical density estimates". In: *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*. Springer. 2013, pp. 160–172.

[86] Sheng Zhou et al. "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions". In: *arXiv preprint arXiv:2206.07579* (2022).

[87] Davoud Moulavi et al. "Density-based clustering validation". In: *Proceedings of the 2014 SIAM international conference on data mining*. SIAM. 2014, pp. 839–847.

[88] Vincent Lostanlen et al. "Energy efficiency is not enough: towards a batteryless internet of sounds". In: *Proceedings of the 16th International Audio Mostly Conference*. 2021, pp. 147–155.

Appendix A

Bambird

Additional functions for the Bambird package were built in order to facilitate the labeling of the data and evaluating the performances of the clustering algorithm used in the Bambird workflow. Performance of the Bambird workflow in correctly classifying the ROIs for each species were evaluated as being labeled "signal" or "noise". Table A.1 presents the evaluation results of the Bambird workflow, where the initial number of True Positive (TP) and False Positive (FP) correspond to the ROIs that were manually labeled. TP and FP respectively represent the number of ROIs labeled by the clustering algorithm as "signal" and "noise". The results of the Precision Initial column indicate the percentage of correct predictions among the positive predictions (i.e. FN / (TP+FN)) and measure the ability of the clustering algorithm not to make mistakes when predicting "signal" versus "noise" for each species.

Species	Number ROIs Initial	Number ROIs Final	TP Initial	FP Initial	TP	FP	TN	FN	Precision Initial
ASIFLA	56	31	20	36	3	28	8	17	36%
ATHCUN	54	23	24	30	0	23	7	24	44%
BUBVIR	61	43	19	42	19	24	18	0	31%
CARPLA	15	9	10	5	8	1	4	2	67%
CHOACU	25	16	17	8	16	0	8	1	68%
CHONAC	58	23	7	51	0	23	28	7	12%
CRYBRE	52	20	47	5	18	2	3	29	90%
CRYCIN	61	33	45	16	33	0	16	12	74%
CRYSOU	23	14	19	4	13	1	3	6	83%
CRYVAR	9	7	8	1	6	1	0	2	89%
DAPATE	48	20	25	23	20	0	23	5	52%
FALCOL	25	6	17	8	1	5	3	16	68%
FALDEI	6	3	2	4	0	3	1	2	33%
FALFEM	47	17	23	24	6	11	13	17	49%
FALPER	18	16	10	8	9	7	1	1	56%
FALRUF	22	12	12	10	11	1	9	1	55%
GLAHAR	38	20	24	14	20	0	14	4	63%
HERCAC	150	32	135	15	32	0	15	103	90%
HYDCLI	3	0	3	0	0	0	0	3	100%
IBYAME	60	44	47	13	42	2	11	5	78%
LOPCRI	81	34	44	37	22	12	25	22	54%
LURSEM	43	19	32	11	17	2	9	15	74%
MEGCHO	16	4	9	7	4	0	7	5	56%
MEGROR	14	6	13	1	6	0	1	7	93%
MEGWAT	30	11	26	4	10	1	3	16	87%
MICGIL	71	8	54	17	7	1	16	47	76%
MICMIR	23	7	18	5	6	1	4	12	78%
MICRUF	97	31	92	5	31	0	5	61	95%
MICSEM	68	51	53	15	49	2	13	4	78%
MILCHI	43	36	15	28	15	21	7	0	35%
NYCAET	7	4	6	1	4	0	1	2	86%
NYCALB	69	46	50	19	42	4	15	8	72%
NYCGRA	47	20	23	24	10	10	14	13	49%
NYCGRI	36	12	9	27	0	12	15	9	25%
NYCLEU	10	2	5	5	1	1	4	4	50%
NYCNIG	20	16	18	2	14	2	0	4	90%
NYCLEU	66	66	65	1	65	1	0	0	98%
PULPER	16	7	11	5	6	1	4	5	69%
STRHUH	83	52	62	21	49	3	18	13	75%
STRVIR	62	19	46	16	17	2	14	29	74%
TINMAJ	114	20	77	37	0	20	17	77	68%

Table A.1: Evaluation of the Bambird workflow for the unsupervised classification of the ROIs as being "signal" or "noise". The number of initial True Positive (TP) and False Positive (FP) correspond to the ROIs manually labeled. TP and FP respectively represent the number of data labeled as "signal" and "noise". The results of the Precision Initial column indicate the percentage of correct predictions among the positive predictions and measure the ability of the clustering algorithm not to make mistakes when predicting "signal" versus "noise" for each species.