UNIVERSITY OF CALIFORNIA SAN DIEGO

An Exploration of Automated Methods for the Efficient Acquisition of Training Data for Acoustic Species Identification

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Electrical Engineering (Machine Learning and Data Science)

by

Jacob Glenn Ayers

Committee in charge:

Professor Curt Schurgers, Chair
Professor Ryan Kastner
Professor Edward Wang

2024

The Thesis of Jacob Glenn Ayers is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

Dedicated to my grandfather, John Harrover Ayers, who helped me build the confidence to pursue my goals but left too soon to witness me achieve them. Umpa, you will forever be missed.

EPIGRAPH

If at first you don't succeed, rewrite your thesis and pretend that was your plan all along.

ChatGPT 3.5

TABLE OF CONTENTS

Thesis Approval Page iii

Dedication iv

Epigraph v

Table of Contents vi

List of Figures viii

List of Tables x

Acknowledgements xi

Vita xiii

Abstract of the Thesis xiv

Chapter 1 Introduction 1
1.1 Training Data Acquisition Challenges 3
1.1.1 Recording Type 3
1.1.2 Label Types: Weak vs. Strong 6
1.1.3 Dataset Types and Challenges Summary 6
1.2 Relevant Datasets 8
1.2.1 Madre de Dios, Peru Audiomoth Deployment 8
1.2.2 La Jolla, California, United States Audiomoth Deployment 9
1.2.3 Xeno-canto 12
1.2.4 Bird Species of Interest 13
1.3 Chapter Acknowledgements 13

Chapter 2 Extracting Training Data from Passive Recordings 16
2.1 Related Works 17
2.1.1 Better Templates 17
2.1.2 Ensemble Learning for False Positive Filtering 18
2.2 Methodology 19
2.3 Zero-normalized Cross Correlation based Template Matching 20
2.3.1 Feature Vector Creation 22
2.3.2 Comparing the Template to the Unknown Signal 22
2.3.3 Clustering Local Scores for Timestamps 23
2.3.4 Statistical Learning Ensemble 24
2.3.5 Ensemble Prediction Aggregation and Sampling 27
2.4 Alternative Verification Methods 27

2.5 Experiments . . . 28
2.5.1 Data . . . 28
2.5.2 Template Matching Setup . . . 28
2.5.3 Statistical Learning Ensemble Setup . . . 30
2.5.4 Evaluation . . . 31
2.6 Results . . . 32
2.7 Chapter Acknowledgements . . . 39

Chapter 3 Extracting Training Data from Purposeful Recordings . . . 40
3.1 Related Works . . . 42
3.1.1 Foreground-background Sound Separation . . . 42
3.1.2 Binary Birdsong Sound Event Detection . . . 44
3.1.3 Template Matching . . . 45
3.2 Methodology . . . 46
3.2.1 Weak to Strong Labeling Pipelines . . . 46
3.2.2 Training Data Length Normalization . . . 51
3.2.3 Multi-species Classification Deep Learning Model . . . 52
3.2.4 Weak Label Aggregation . . . 53
3.3 Experiments . . . 54
3.3.1 Human Strongly Labeled Data . . . 54
3.3.2 Generating Temporal Indicator Vectors with each WTS Pipelines . . . 55
3.3.3 Evaluating Weak to Strong Label Pipelines . . . 57
3.3.4 Efficientnet Training . . . 57
3.3.5 Efficientnet Evaluation . . . 58
3.4 Results . . . 59
3.5 Chapter Acknowledgements . . . 61

Chapter 4 Conclusion . . . 65
4.1 Contributions . . . 65
4.2 Discussion . . . 67

Bibliography . . . 69

LIST OF FIGURES

Figure 1.1. Passive Audio Example from Madre de Dios, Peru of a Screaming Piha Vocalization. Visualized using the web-based audio labeling platform, Pyrenote. 4

Figure 1.2. Purposeful Audio Example from Xeno-canto (XC-272411) of Screaming Piha Vocalizations. 5

Figure 1.3. Comparison of Weak Labels (top) and Strong Labels (bottom) on Screaming Piha Audio (XC-272411) 6

Figure 1.4. Madre de Dios, Peru Audiomoth Deployment June-September 2019 9

Figure 1.5. Madre de Dios, Peru Audiomoth Deployment Tree Attachment Example . 10

Figure 1.6. Scripps Coastal Reserve - La Jolla, California, United States Audiomoth Deployment August 2021 11

Figure 1.7. Scripps Coastal Reserve Audiomoth Deployment - Device Housing and Attachment. 11

Figure 1.8. Xeno-canto Taxonomy Count Screenshot Taken 05/28/2024 12

Figure 1.9. Images of Relevant Bird Species. 13

Figure 1.10. Spectrograms of Bird Species Vocalizations. 14

Figure 2.1. Template Matching Verification with Ensemble Learning Flowchart 21

Figure 2.2. Screaming Piha Vocalization Autocorrelation. Correlation outputs (top) Vocalization (bottom) 23

Figure 2.3. Screaming Piha Vocalization Cross Correlation. Correlation outputs (top) Unknown Signal (bottom) 24

Figure 2.4. Screaming Piha ZNCC Timestamp (yellow) Creation based on Threshold (red) 25

Figure 2.5. Example Kaleidoscope Lite Interface for Template Matching Verification . 29

Figure 2.6. Purposeful and Passive Recording Templates 30

Figure 2.7. Template Matching Confidence Histograms 35

Figure 2.8. Simulated Template Matching Verification Runs 36

Figure 2.9. AUC Ratio Side-by-side Bar Charts 38

Figure 3.1. Example of Foreground-background (green-red) concatenation from Sprengel et al, 2016 43

Figure 3.2. Weak-to-strong (WTS) Pipeline Flowchart 47

Figure 3.3. Sprengel et al. 2016 Foreground-background Separation Demonstration . . 49

Figure 3.4. Pyrenote Interface for Strong Labeling on a Bright-rumped Attila Xenocanto clip 55

Figure 3.5. EfficientNet F1 Metrics Side-by-side Bar Charts 63

LIST OF TABLES

Table 1.1. Dataset Categorization Table 7

Table 2.1. Parameters Used for Template Matching and Ensemble Learning 34

Table 2.2. Template Matching Verification Summary 37

Table 3.1. Weak to Strong Labeling Pipeline Training Estimation Summary 59

Table 3.2. Performance Metrics of EfficientNet Models on the Weak Label Prediction Task 62

ACKNOWLEDGEMENTS

If I had spoken to myself during my high school years when I was in hot pursuit of becoming a screenwriter, I imagine my younger self would be dumbfounded to know that I have spent eight years pursuing a master's degree in electrical engineering. To those that have helped guide me on what often felt like a Sisyphean task, you have my deepest gratitude.

I must first thank some of the fantastic Professors from my Santa Barbara City College days including Jeffrey Gray, my first professor that helped me garner momentum at the onset of my pivot into STEM. I must acknowledge Justina Bueller for her wonderful Environmental History course that set me on a path towards research with an emphasis in sustainability. I owe Kira Minkova and Stephen Strenn a great deal of gratitude for helping me build a foundation in computer science that prepared me for this work.

Naturally, thanks are owed to the wonderful Principal Investigators of my work at Engineers for Exploration, Curt Schurgers and Ryan Kastner. Without their willingness to put the Acoustic Species Identification project under the leadership of a recent transfer student without prior research experience and little to no programming experience at the onset of the Covid-19 pandemic, I can only imagine how much more challenging navigating career opportunities would be.

Without the help of the dozens of students that have made contributions to this project since its inception in the Winter of 2020 including Erika Joun, Gabriel Steinberg, Yoo-Jin Hwang, and Mugen Blue as well as my co-leads Sean Perry and Samantha Prestrelski that picked up my slack when life forced me to attend to different obligations. Furthermore, I want to acknowledge Tianqi Zhang and Ludwig Von Schoenfeldt who are taking on the mantle of responsibility of the project for future generations and give me the confidence to sit in the shade of my own vine and fig tree as the work outlives me when I am gone.

Finally, I must thank my parents, Mark Ayers and Sandra Renshaw, for supporting me through these college years that they never quite had the opportunity to pursue themselves. I want to thank my little sister Francesca for being the best little sister a big brother could ever ask

for and I will work to the best of my ability to afford you the opportunities I have had in life.

Chapters 1 and 3 contains screenshots from our team's manual audio labeling software, Pyrenote. Sean Perry was the lead developer of the software. The thesis author helped delegate resources and advised the development.

Chapter 1 describes an Audiomoth deployment in La Jolla, California. Chapter 2 describes the use of data from said deployment. The thesis author led this Audiomoth deployment that was funded by Engineers for Exploration under the leadership of Curt Schurgers and Ryan Kastner.

Chapter 1 describes an Audiomoth deployment in Madre de Dios Peru. Chapters 2 and 3 describe the use of data from said deployment. The audio data collected from Peru was kindly provided to us by our San Diego Zoo Wildlife Alliance collaborators Mathias Tobler and Ian Ingram.

Chapter 2 and Chapter 3 describe the use of an algorithm to cluster temporal predictions into timestamps. This algorithm has yet to appear in a publication. It was developed by Gabriel Steinberg.

Chapter 3 describes a bird sound event detection neural network called TweetyNet. The thesis author summarizes the training of the network as well as its application of estimating human training data. Our team has been using this model for a while, but it has not yet appeared in any publications. Mugen Blue was the team member that trained the TweetyNet weights used in this work.

Chapter 3 describes training the Efficientnet neural network architecture. To accomplish this, the thesis author used our team's multiclass training pipeline that has yet to be used in a publication. This pipeline was developed primarily by Sean Perry and Samantha Prestrelski.

VITA

2018	Tutor of Mathematics, Santa Barbara City College
2019	Associates of Science in Engineering, Santa Barbara City College
2021–2024	Tutor/Teaching Assistant, Electrical and Computer Engineering Department, University of California San Diego
2022	Bachelor of Science in Electrical Engineering, University of California San Diego
2023	Autonomous Systems Engineering Internship, Nokia Bell Labs
2024	Master of Science in Electrical Engineering, University of California San Diego

PUBLICATIONS

"Challenges in Applying Audio Classification Models to Datasets Containing Crucial Biodiversity Information." Climate Change AI Workshop, International Conference on Machine Learning (ICML). 2021.

"Reducing the Barriers of Acquiring Ground-truth from Biodiversity Rich Audio Datasets Using Intelligent Sampling Techniques." Tackling Climate Change with AI Workshop, Conference on Neural Information Processing Systems. 2021.

"Pyrenote: a web-based, manual annotation tool for passive acoustic monitoring." 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2021.

ABSTRACT OF THE THESIS

An Exploration of Automated Methods for the
Efficient Acquisition of Training Data for Acoustic Species Identification

by

Jacob Glenn Ayers

Master of Science in Electrical Engineering (Machine Learning and Data Science)

University of California San Diego, 2024

Professor Curt Schurgers, Chair

Passive acoustic monitoring is a field that strives to understand the health of ecosystems around the world through the acoustics of natural soundscapes. By identifying fauna vocalizations within soundscapes, we begin to build a quantitative understanding of local biodiversity populations, a key indicator of ecosystem health. The reduced cost of audio recorders have enabled researchers to collect datasets at a scale untenable in years past. These datasets are too vast for exhaustive human identification of species vocalizations. To which, researchers hope to train deep learning models for automated acoustic species identification to mitigate the burden of human labor.

To build robust deep learning models that can differentiate between species vocalizations, one must have access to a large amount of training data. The availability of labeled audio data is a bottleneck to training such models. To acquire labeled training data, researchers must develop techniques to extract vocalizations from their collected recordings. Alternatively, there exists audio data that is available to the scientific community where species have been identified in a clip (weakly labeled), but not where (strongly labeled), requiring extra processing to identify where relevant sounds occur to be adequate training data.

This thesis demonstrates methodologies that mitigate human effort in training data acquisition. For unlabeled soundscape recordings we explore how best to acquire vocalizations of a species of interest assuming you have identified at least one recording. For weakly labeled recordings that have been made available to the scientific community, we explore methods that automate the extraction of relevant sounds.

Chapter 1

Introduction

There is an ever-growing repertoire of evidence to support the notion that human activity is the leading cause of biodiversity loss around the globe [27] [73]. With that comes an increase in the number of efforts to alleviate biodiversity loss and methods to measure the efficacy of such efforts [54]. We will focus on methods related to fauna monitoring. Some historical methods for surveying fauna populations include feeding site monitoring [22], catch and release [74] [45], and track monitoring [38]. These methods have historically required on-site monitoring by researchers that is time consuming and the data can be restricted to smaller localities [20]. There have also been increased restrictions on monitoring techniques that can be harmful to individual specimens, such as catch and release methods [48]. These challenges in on-site surveys of limited areas that are time consuming and potentially harmful to fauna specimens being studied culminate in major blockers for large-scale biodiversity surveys.

To face these scaling challenges, scientists have turned to remote-sensing technologies such as satellites, camera traps, and audio recorders [51] [71] [69] [63] [64]. Such technology helps reduce the cost of data collection and have made the data more widely available to the scientific community [61]. This reduction of human effort in collecting data in turn creates another problem of extracting information from a potentially overwhelmingly large dataset. To address this big data problem, scientists have turned to machine learning algorithms that are trained on data and used to make inferences on similar data without an explicit set of instructions

[59]. The most robust of these algorithms come from the field of deep learning [52] [42] [26]. Deep learning models function as universal approximators that capture complex relationships within datapoints and their corresponding labels [15]. To achieve this in intricate tasks like image classification, these algorithms necessitate substantial amounts of training data, which is crucial for learning the nuanced features and patterns required for accurate predictions [17]. This need for large datasets stems from the high dimensionality and complexity of the input spaces they operate in, ensuring the models generalize well across varied data distributions and real-world scenarios. The expensive hardware requirements to train effective large image classification models from scratch can often be out of reach for many researchers [75]. Fortunately, a large breadth of foundational models that have already been trained for image classification are available to the scientific community [44] [32]. These models are commonly leveraged for training on smaller datasets for specific tasks in a process called transfer learning, which lowers the barrier of entry for deep learning applications [56]. Foundational deep learning models have been successfully trained using transfer learning methods to help conduct biodiversity surveys on large camera trap datasets [59] [66].

Image capture is excellent for larger fauna but the limits become immediately apparent as one wishes to study population health of smaller fauna such as frogs, insects, bats, and birds. Species from these groups often serve as indicator species on the health of a natural environment [21] [33] [8] [67]. To this end, scientists hope to leverage the fact that many of these species produce discernible vocalizations that can be recorded [14] [45] [47] [46]. To gather audio clips, they utilize microcontroller-based audio recorders designed for periodic, long-term soundscape recording [28]. As is the case with many remote sensing technologies, these passive acoustic monitoring datasets are often too large making it infeasible for humans to listen to in their entirety[30]. To reduce the amount of audio to be listened to by humans, researchers hope to use machine learning models trained to identify relevant sounds on audio data. With that being said, there is not as much work done in developing machine learning models that make inferences on audio signals directly. To which, researchers aim to use image classification based foundational

deep learning models that have yielded success in similar works with camera traps [40]. The use of image classification models on audio data can be justified by using techniques such as the Short-time Fourier Transform (STFT) [23] that creates an image called a spectrogram that can be used for training and inference on deep learning models [36] [62] [14]. A few examples of spectrogram feature vectors derived from the STFT can be seen in Figure 1.10.

With this transformation of audio into the image domain we can leverage deep learning models designed for image classification. Transfer learning techniques can lower the barriers of entry of training deep learning models for acoustic species identification [58] [40] [36]. Despite the reduced data requirements to train models with transfer learning, the collection of adequate labeled training data of species vocalizations is a non-trivial task. The remainder of this chapter will describe these challenges in depth with respect to bird species vocalizations. Furthermore, the remaining chapters will present methodologies to tackle these challenges.

1.1 Training Data Acquisition Challenges

This section will highlight how the general approach to recording sounds will fundamentally impact the ease of acquisition of labeled training data of species vocalizations. It will also include a discussion on what kind of labels are satisfactory for training neural networks.

1.1.1 Recording Type

Oftentimes when researchers wish to acquire labeled training data, they can look into literature that may highlight certain methods on a certain audio dataset that may not end up translating well to their own recordings. To highlight this challenge, this section will distinguish between approaches to recording audio data that influences the ease of acquisition of labeled training data.

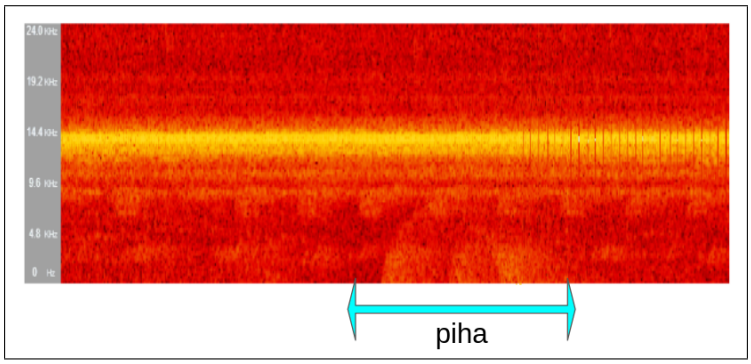


Figure 1.1. Passive Audio Example from Madre de Dios, Peru of a Screaming Piha Vocalization. Visualized using the web-based audio labeling platform, Pyrenote.

1.1.1.1 Passive Recordings

We borrow from the term passive acoustic monitoring to define a category of audio recordings that are typically collected with low-cost audio recorders such as Audiomoths [28] in remote environments [41]. These recorders are typically set to record on periodic intervals for long periods of time. That is to say that once all of the data is collected there is likely to be plenty of audio that fails to contain information of direct value to the intent of the original deployment. Furthermore, as bird species vocalize in the environment, even audio that does contain vocalizations from a species of interest can be quite challenging to identify depending on the distance from the recorder. One such example of a Screaming Piha (*Lipaugus vociferans*) can be seen in Figure 1.1 where the birdsong blends in with the background.

In section 1.2 we will discuss two Audiomoth deployments used in this study that fall into this category of passive recordings.

1.1.1.2 Purposeful Recordings

We distinguish another category of recordings that are often collected with more purpose compared to the aforementioned passive methods. For example, if a person is walking around a natural environment with audio equipment and hears a bird vocalization of interest, they may stop and begin to record an audio clip. Now, in this example, by reacting to a sound and beginning to record, the resulting audio clip is likely to contain a vocalization that is easily distinguishable

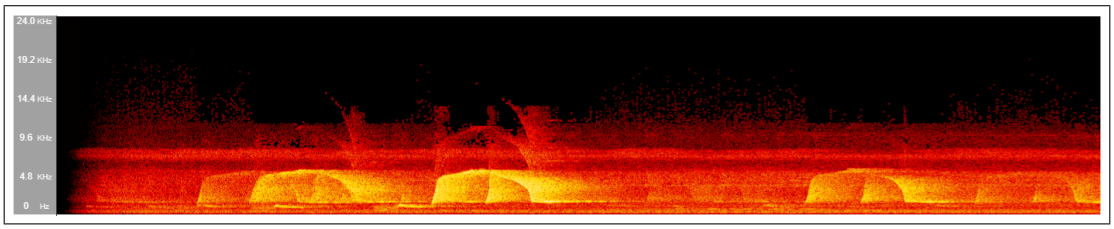


Figure 1.2. Purposeful Audio Example from Xeno-canto (XC-272411) of Screaming Piha Vocalizations.

from the background noise of the environment. The recording can potentially be focused on the subject sound if the individual recording has directional equipment. Furthermore, in this example, the individual will likely stop recording once the sounds of interest cease. In this case, the individual has produced an audio recording that has clear sounds of interest from a species as well as clean start and stop times that mitigate the inclusion of irrelevant sounds. This culminates in an audio clip where the regions best fit for training are clearer, and are likely easier for a machine learning model to classify during inference time. An example clip we would classify as purposeful can be seen in Figure 1.2 where Screaming Piha calls are the most distinct sound in the recording and the clip stops when the calls stop.

When more purposeful audio recordings are collected, they are more likely to be tagged with information. For instance, if the individual recording is knowledgeable of the species being recorded, either by sight or sound, they may include that information with the recording.

However, it is not always the case that the purposeful recording is tagged with species information. For example, the person making the recording may not be an expert or is unfamiliar with a particular species being recorded. In this case, they are still recording with a purpose, despite having less precise information associated with the recording.

To summarize, purposeful recordings are made with the intent of recording distinct vocalizations, and often, but not always, contain precise information related to species or taxonomy.

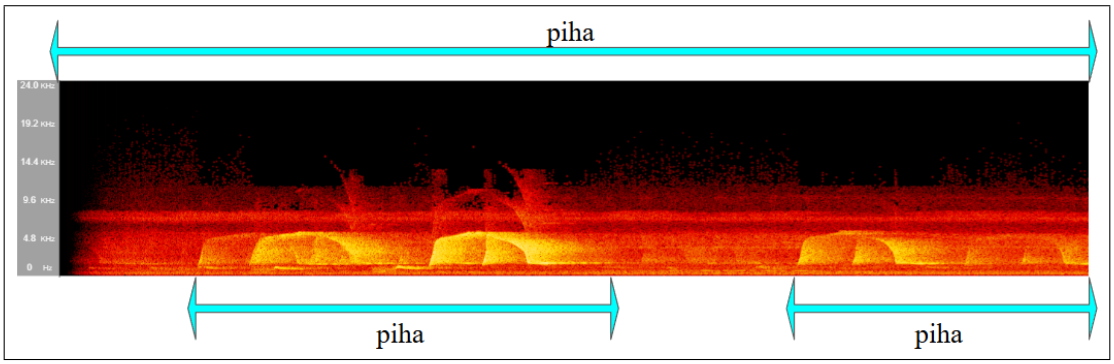


Figure 1.3. Comparison of Weak Labels (top) and Strong Labels (bottom) on Screaming Piha Audio (XC-272411)

1.1.2 Label Types: Weak vs. Strong

This section will detail what it means to label an audio clip. In the field of machine learning, when data is referred to as labeled, this generally means that you have what we call a strong label where the class of interest is clearly defined within the datapoint. As alluded to in the prior section, if an audio clip is recorded and tagged with some information with respect to the entire clip, it becomes challenging to define labeled training data. That is because the exact timestamps when the sound of interest occurred is not included. We refer to labels defined across an entire clip as a weak label. An example of the difference between a weak and strong label on a clip containing Screaming Piha vocalizations can be seen in Figure 1.3.

Attempting to train deep learning models with weakly labeled data can be dangerous as there is an inherent risk of including sounds in the training process that do not correspond to the weak label. Instead, it is desirable to train deep learning models with strong labels instead, which are very much focused on only the signals of interest.

1.1.3 Dataset Types and Challenges Summary

We consolidate the types of data we will focus on based on the previously discussed recording categories and what information is typically available with them in Table 1.1. We mark passive recordings with weak labels with an ”X” since, the assumption of a passive recording is

Table 1.1. Dataset Categorization Table

	Weak Label	No Weak Label
Purposeful	Type 1	Type 2
Passive	X	Type 3

that it is unprocessed and consequently has no weak label.

With Table 1.1 in mind, we can highlight the challenges associated with each of these dataset types. Type 3 recordings are characterized by long stretches of recordings that are often sparse of vocalizations of interest. Furthermore, these vocalizations can often blend in with background noise. To add to the challenge, upon collection, no labels are available. To which, any technique that attempts to streamline the human training data verification process will have to filter out long stretches of environmental noise and pick out audio segments likely to contain the species of interest. Purposeful recordings are generally easier to extract labeled training data from. As discussed, these audio clips have been recorded with the intent to record sounds that happen to be relevant to the machine learning goals. In the case where we do not know the weak label species of purposeful recordings (Type 2), we do know that the data will be skewed towards the taxon of interest. So in these cases there will be much less irrelevant noise compared to passive recordings. However, the challenge of separating the species of interest from species of the same taxon remains. The easiest type of data to extract strongly labeled training data from are Type 3 recordings. In these cases, we know that the species of interest exist in the audio clips and they tend to be easy to distinguish from irrelevant background noise as they were purposefully recorded generally in the foreground. There still exists the risk of focal sounds that are not the weakly labeled species, but those are less prevalent compared to the other data types.

The contributions of this thesis will be split into two Chapters. Chapter 2 will focus on methodologies related to extracting labeled training data from Type 3 recordings. The chapter will also include results of the methodologies applied to Type 1 and Type 2 recordings as well for the sake of comparing and contrasting approaches to recording types. Chapter 3 will focus

on extracting strongly labeled data from Type 1 recordings. These strongly labeled datasets will be used to train neural networks that are evaluated across all 3 types of recordings. Extracting training data from Type 2 recordings is de-emphasized compared to the other types as they are less commonly used to try and collect training data for deep learning models.

1.2 Relevant Datasets

This section will cover the specific passive and purposeful datasets that will be used for experiments in the remainder of the thesis. We will also explicitly state which bird species we will seek labeled training data for.

1.2.1 Madre de Dios, Peru Audiomoth Deployment

The first passive dataset, as defined in section 1.1.1.1, relevant to this thesis was collected in the state of Madre de Dios, Peru in the Amazon Rainforest. The Amazon Rainforest is one of the greatest sources of biodiversity in the world [2]. Its dense vegetation is critical for stabilizing carbon dioxide levels that contribute to climate change [7]. Unfortunately, recent decades have shown that the Amazon has greatly degraded over the years due to deforestation from sources such as illegal logging for cattle farms and forest fires [55]. Organizations such as the Forest Stewardship Council (FSC) provide certifications for corporations seeking to sustainably harvest timber from natural environments such as the Amazon Rainforest [5]. The San Diego Zoo Wildlife Alliance (SDZWA) performed an Audiomoth deployment along two such FSC certified logging concessions run by Forestal Otorongo and MADERACRE. This deployment was carried out with the hopes of translating successes in monitoring the jaguar population in the area with camera traps using deep learning [66] into monitoring smaller species with Audiomoths. In total, 35 Audiomoths were deployed alongside logging roads or in unlogged forest. The deployment took place over June to September of 2019. A Google Maps screenshot of the deployment can be seen in Figure 1.4. The device housings were attached to trees as show in Figure 1.5. The Audiomoths were programmed to periodically record 1 minute of audio every 10 minutes at a

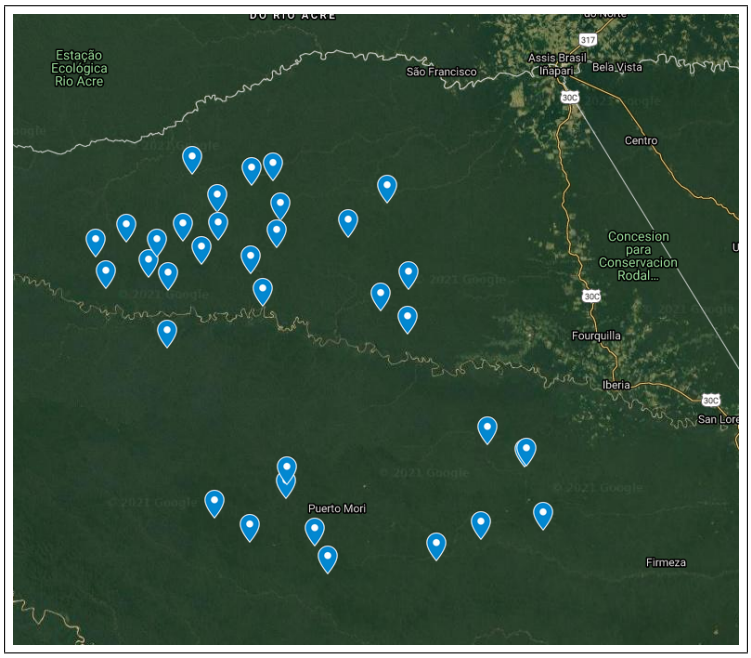


Figure 1.4. Madre de Dios, Peru Audiomoth Deployment June-September 2019

384 kilohertz sampling rate. In total, 31 devices successfully collected about 4 terabytes of data amounting to about 1500 hours of audio.

This dataset will be used to explore approaches to extract labeled training data of bird species vocalizations from passive recordings in Chapter 2. Furthermore, in Chapter 3, we will explore how best to extract training data from weakly labeled, purposeful recordings, and train models on said data. Those models will be evaluated on recordings from this dataset

1.2.2 La Jolla, California, United States Audiomoth Deployment

California coastal sage scrub chaparral environments have been in decline in recent decades [6]. Two of the main contributors are invasive species and anthropogenic nitrogen deposition [1].

We performed an Audiomoth deployment inside the Scripps Coastal Reserve (SCR) in La Jolla, California. The deployment served as an opportunity for our team to garner expertise in performing Audiomoth surveys. It also gave us the opportunity to meet local birding enthusiasts and garner their feedback on our team's audio labeling system Pyrenote [57]. The SCR is

Figure 1.5. Madre de Dios, Peru Audiomoth Deployment Tree Attachment Example

managed by the UC San Diego Natural Reserve System. This environment contains over 100 bird species. From August 10th to the 24th, 2021 10 Audiomoths were deployed recording 1 minute every 10 minutes at a 384 kilohertz sampling rate. During this time, the SCR was not open to the public due to the global Covid-19 pandemic. The Audiomoths were housed in either the standard-issue Audiomoth housing or in Ziploc bags. The devices were attached to coastal sage scrubs such as Lemonade berry (*Rhus integrifolia*) bushes and California sagebrush (*Artemisia californica*).

Figure 1.6 shows where on the SCR the Audiomoths were placed. Figure 1.7 shows how the Audiomoths were attached to different california coastal sage scrub species.

This dataset will be used to explore approaches to extract labeled training data of bird species vocalizations from passive recordings in Chapter 2. However, in Chapter 3, when we explore how best to extract training data from weakly labeled, purposeful recordings, we will not use audio clips from this dataset, instead focusing our energy on the Madre de Dios, Peru dataset.

Figure 1.6. Scripps Coastal Reserve - La Jolla, California, United States Audiomoth Deployment August 2021

(a) Open Acoustic Devices housing on Lemonade berry (*Rhus integrifolia*).

(b) Ziploc bag housing on California sagebrush (*Artemisia californica*).

Figure 1.7. Scripps Coastal Reserve Audiomoth Deployment - Device Housing and Attachment.

Figure 1.8. Xeno-canto Taxonomy Count Screenshot Taken 05/28/2024

1.2.3 Xeno-canto

With respect to bird species vocalizations, the most widely used database of purposeful recordings, defined in section 1.1.1.2, is called Xeno-canto. An example of a Screaming Piha Xeno-canto audio clip can be seen in Figure 1.2.

Xeno-canto was built with the intent of providing the public with sound recordings of all wildlife. Consequently making it very easy to pull data from using warbleR [3] and its widespread adoption in the ornithology community leading to a plethora of recordings. Figure 1.8 shows how many recordings they host by taxonomy. Furthermore, most of the audio is labeled for the presence of certain species and they have many features such as the ability to select clips based on region.

Though the focus of Chapter 2 will be on methods to extract labeled training data of bird species vocalizations from passive recordings, we will apply similar the same methods to Xeno-canto purposeful recordings. We do this to be thorough in our comparisons between purposeful and passive recordings. Chapter 3 will focus on methods for extracting training data from purposeful recordings with weak labels. All of the relevant techniques to accomplish this will be applied to data collected from Xeno-canto. Furthermore, when we go to evaluate deep learning models trained on these labeled training datasets, we will evaluate the models on both passive recordings from Madre de Dios and purposeful recordings from Xeno-canto to be thorough in our comparisons on the basis of recording types.

Figure 1.9. Images of Relevant Bird Species.

1.2.4 Bird Species of Interest

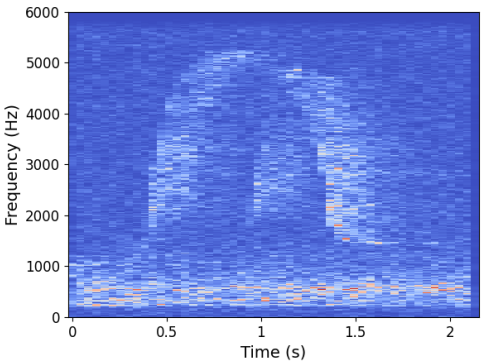
If the goal is to collect training data of bird species vocalizations, we must define which species we are interested in collecting data for. The first species of interest is the Screaming Piha (*Lipaugus vociferans*) that is iconic to the Amazon due to its powerful ”Pee-haw” sound, making it one of the loudest bird species in the world [53]. The second Amazonian bird species we will investigate is the Bright-rumped Attila (*Attila spadiceus*) known for its distinct red iris and its melodic ”whew-whit” dawn song [43]. The final species is the Common Poorwill (*Phalaenoptilus nuttallii*) that ranges throughout the Western United States and into Mexico. It is a noctural species that has an almost ethereal, low-frequency ”poor-will” mating call.

Images of all of the species discussed are shown in Figure 1.9. Spectrograms of said species can be seen in Figure 1.10.

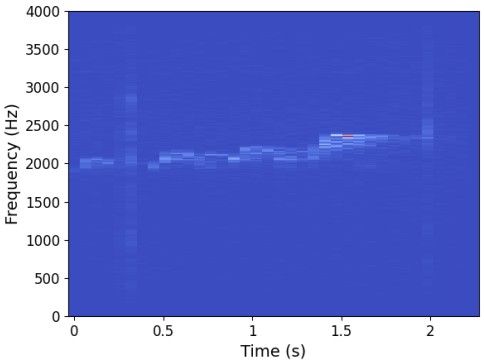
1.3 Chapter Acknowledgements

Chapter 1 contains screenshots from our team’s manual audio labeling software, Pyrenote. Sean Perry was the lead developer of the software. The thesis author helped delegate resources and advised the development.

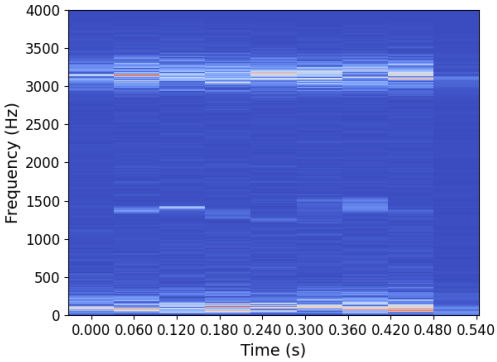
Chapter 1 describes an Audiomoth deployment in La Jolla, California. The thesis author led this Audiomoth deployment that was funded by Engineers for Exploration under the leadership of Curt Schurgers and Ryan Kastner.



(a) Screaming Piha (*Lipaugus vociferans*)



(b) Bright-rumped Attila (*Attila spadiceus*)



(c) Common Poorwill (*Phalaenoptilus nuttallii*)

Figure 1.10. Spectrograms of Bird Species Vocalizations.

Chapter 1 describes an Audiomoth deployment in Madre de Dios Peru. The audio data collected from Peru was kindly provided to us by our San Diego Zoo Wildlife Alliance collaborators Mathias Tobler and Ian Ingram.

Chapter 2

Extracting Training Data from Passive Recordings

In chapter 1, we defined passive audio recordings as datasets characterized by recordings collected periodically that tend to contain long stretches of audio that is irrelevant to species vocalizations of interest. Furthermore, the species vocalizations of interest have a tendency to be challenging to distinguish from other environmental soundscape audio. When such passively recorded datasets are collected with the intent to identify bird species vocalizations using deep learning, the initial steps of gathering training data for the models present a non-trivial challenge of mitigating human labor hours identifying relevant species vocalizations that usually requires listening to hours of irrelevant audio. In this chapter, we demonstrate a process that we view as the lowest barrier of entry for narrowing down audio recordings to be verified by humans: template matching with species vocalizations of interest.

The term template matching encapsulates a set of methods that involve taking a known signal of interest, referred to as a template, and using it to find similar regions within an unknown signal. Some applications of this include channel estimation and system synchronization with preamble signals in modulator-demodulator devices [25] and object detection in images [9]. For the field of passive acoustic monitoring, template matching offers an accessible and efficient method for initial data parsing, allowing researchers to sift through vast amounts of audio data and pinpoint specific bird calls without the need for extensive pre-labeling or complex algorithms.

One of the go-to software libraries for template matching with animal vocalizations is called monitoR that encapsulates a popular binary template matching technique [68].

When a large dataset is collected from a noisy environment and a bird species vocalization template is used in a template matching run to identify similar sounds (true positives (TPs)), researchers often end up with template-matched audio that does not contain the vocalization if interest (false positives (FPs)). So, say there are a total of N audio clips that have been produced from a template matching run. In the N clips there are M TPs and N-M FPs. Now, say we can only afford to have humans label n clips, where n is less than or equal to N. In the n clips, there are m TPs and n-m FPs. If we can minimize n-m, this would mean that we are maximizing the labeled training data TP yield given a constraint on the number of clips that we label. This raises the critical question of how to best select template-matched audio given to humans to verify that minimize the number of FPs they listen to.

In the remainder of the chapter, we will explore various approaches and methodologies to address this challenge of minimizing the number of FPs labeled by humans.

2.1 Related Works

In this section we summarize work of other researchers in addressing the challenge of template matching verification. We will also talk about how our work differs from theirs, though we will describe our methodologies and experiments in more detail later in this chapter.

2.1.1 Better Templates

When considering the best methods to yield the most effective template matching run, a natural place to look is to seek the best template. When it comes to template matching with audio data, much work has been done in the space of identifying the most prominent regions of image representations of sound, spectrograms [35]. There are many approaches to building spectrogram representations of sound, including the linear Short-Time Fourier Transform (STFT) and logarithmic scales. Notably, the use of the Mel Scale and Mel-frequency cepstral coefficients,

which attempt to mimic human auditory perception and provide robust features for tasks like speech and bird song recognition, has shown promise [29]. Furthermore, the monitoR library allows users to make fine-grain manipulations to the time and frequency regions of a template [37].

We recognize that determining the best templates a-priori is challenging due to the variability in recording conditions and species vocalizations. Consequently, rather than extensively optimizing template selection, we focus on filtering false positives, assuming a reasonable form of template matching was employed. This approach allows us to maximize the effectiveness of our template matching runs while managing the inherent uncertainties of passive recordings.

2.1.2 Ensemble Learning for False Positive Filtering

Our work related to maximizing the value of template matching verification presented in this thesis builds on top of the work done by Balantic [4]. In this paper, study bird species within the Sonoran Desert in California. In their work they use the monitoR library's binary template matching capabilities to identify templates of species of interest within the Sonoran Desert in California. Their template matching runs lead to template-matched audio that contains TPs and FPs. To filter out the FPs they develop machine learning models. They first label a subset of their overall template-matched audio to collect some examples of TPs and FPs. The audio from this human verification is used to train and evaluate several statistical learning models that are well-suited for smaller training datasets. They use these trained models to build an ensemble wherein all models are used to make predictions on audio clips weighted by their performance on a validation dataset. The ensemble is then used to inference on the remaining unlabeled template-matched audio leading to less FPs overall.

Our work is distinguished from theirs as we are working with different species in different ecosystems. Our work expands beyond the use-case of soundscapes by comparing results on passive recordings as well as purposeful recordings. We use a different form of template matching that involves cross-correlation. We chose a correlation based approach for easier interpretability.

We use the same statistical learning models that we will discuss in the methodologies. When we use the ensemble models to determine which of the remaining unlabeled audio pool should be labeled, since we don't independently evaluate each model, we weigh their predictions uniformly. We set up an experimental design with an upper and lower bound for measuring the percentage of true positive identifications that are found given what percentage of the pool has been labeled. This requires us to have verified a template matching run in its entirety to run such simulations. Furthermore, with this experimental setup we can make the valuable contribution of performing a form of ablation study to compare the ensemble approach to more naïve approaches such as random sampling and sampling template matching outputs with the highest correlation to the template used. Our experimental setup enables us to quantitatively compare these approaches to an ideal upper bound.

2.2 Methodology

To tackle the challenge of efficiently acquiring training data directly from passive recordings collected from natural soundscapes, we borrow methods from the field of active learning. Active learning in the context of machine learning is a description of a set of problems wherein there is a large collection of datapoints with unknown labels that have a certain cost to label, and the goal is to find a classifier that maps the datapoints to their respective labels with a minimal cost [16]. In our case, we are not necessarily looking for the optimal classifier that labels all of the datapoints, rather we wish to maximize the number of datapoints observed of a certain class in a large unlabeled pool given a restriction to the number of datapoints a human can verify.

An overview of our approach to to this challenge via a two stage template matching verification is as follows -

1. Perform template matching to reduce the recordings down to those with similarities to the template species of interest.

2. Determine the value of n, i.e., clips you will verify.

3. Label a subset of 0.2*n from the template-matched audio. A 20-80 split is a common data division in machine learning.

4. Use the labeled subset to train several statistical learning models that are effective in training data scarce settings. The models will be trained as binary classifiers to distinguish between true positives and false positives.

5. Combine the models into an ensemble, a process wherein all of the models vote on the presence of the species of interest.

6. Use the ensemble to inference on the remaining unlabeled template-matched audio. Each datapoint gets an integer value [0, k] where k is the number of statistical learning models in the ensemble.

7. Order the remaining unlabeled template-matched audio according to the ensemble presence vote.

8. Select and label remaining 0.8*n audio clips based on presence vote in descending order

This overview can be seen in flowchart form in Figure 2.1. The remainder of the section will provide more details with respect to each of these steps. Furthermore, smaller details regarding parameter choices that are necessary to reproduce this work, will be covered in the experiments.

2.3 Zero-normalized Cross Correlation based Template Matching

When performing template matching on audio data, several fundamental steps must be considered. First, one must decide how to represent the audio data that contains the template, what we will call a feature vector. Next, the method for comparing the template to unknown signals needs to be determined. Finally, the results of these comparisons must be clustered into

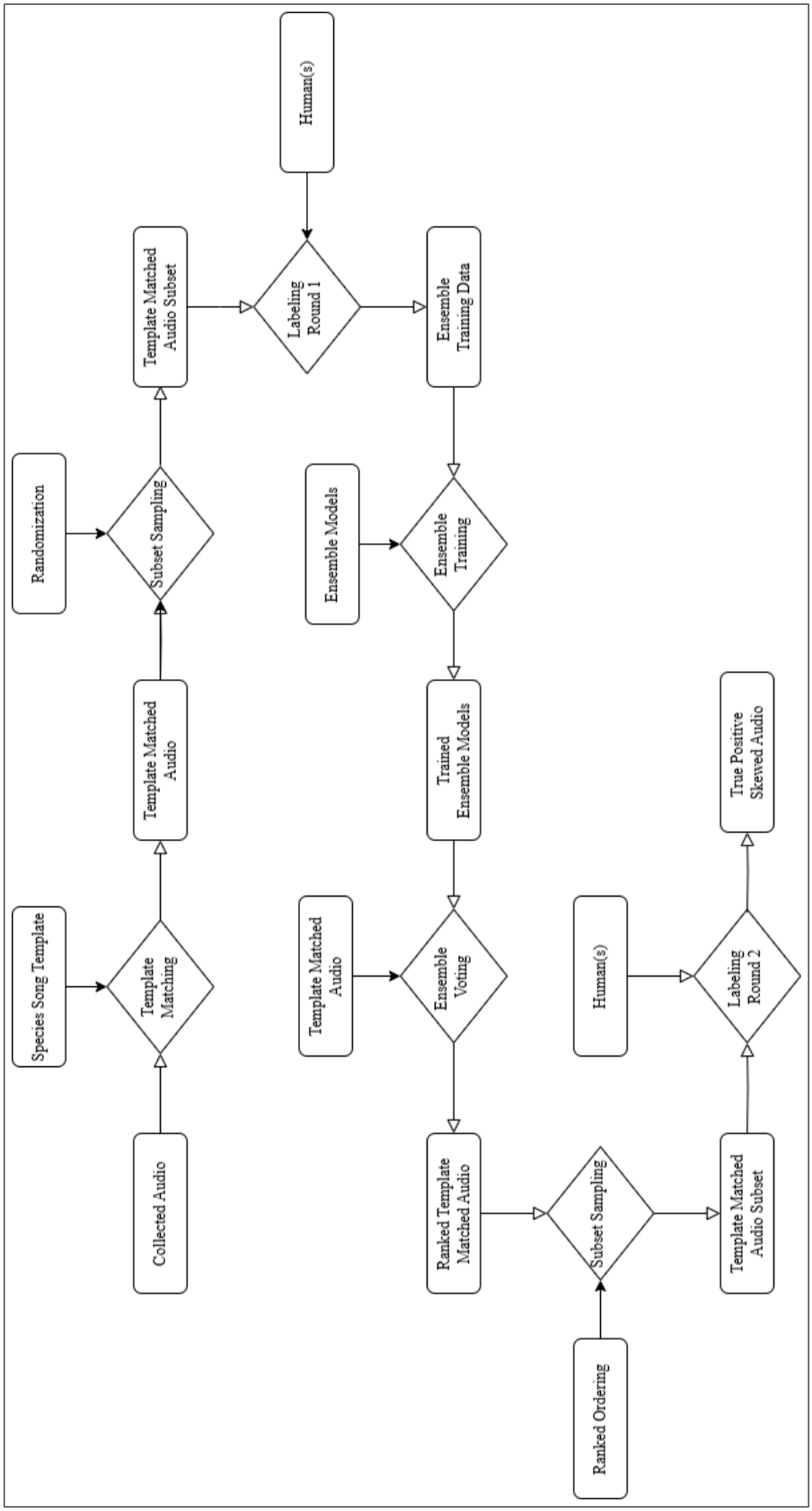


Figure 2.1. Template Matching Verification with Ensemble Learning Flowchart

timestamps. This section will detail each of these steps, including the pre-processing required for template representation, the comparison techniques used, and the clustering methods for accurate timestamp identification.

2.3.1 Feature Vector Creation

In our experiments, we aim to create a sound representation feature vector that emphasizes the vocalization frequency ranges of interest, ensuring that bird songs are captured accurately while irrelevant sounds are minimized. To achieve this, we first downsample the audio signal to a sample rate that is low enough to remove irrelevant high-frequency sounds, but high enough to retain all the birdsong frequencies of interest. Next, we apply a Butterworth bandpass filter [60] to further exclude frequencies outside the vocalization range, thereby focusing our analysis on the relevant audio spectrum. We then compute the magnitude component of the Short-Time Fourier Transform (STFT) using a Hanning window, normalizing the resulting values by dividing by the maximum power to scale the range to [0,1]. This preprocessing ensures that our feature vector is optimized for detecting the bird vocalizations while reducing the influence of irrelevant noise.

2.3.2 Comparing the Template to the Unknown Signal

Zero-normalized Cross-correlation (ZNCC) is a useful method for comparing a template to an unknown signal. It functions similarly to the Pearson Correlation Coefficient, providing outputs normalized over the range [-1, 1], where -1 indicates perfect anti-correlation and 1 indicates perfect correlation [12]. When a template is compared to itself, this process is known as autocorrelation and will yield a perfect correlation of 1. An example of this is shown in Figure 2.2

To compute the ZNCC between a template and an unknown window, all the feature vector creation steps described in section 2.2.2 are first applied to the unknown window. From there, the mean and standard deviations of both the template and unknown window are computed.

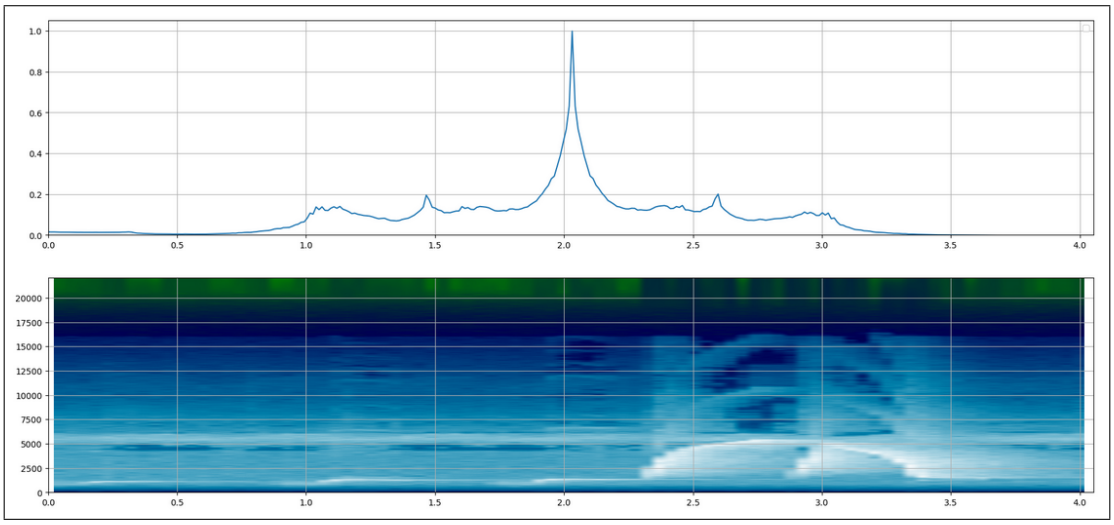


Figure 2.2. Screaming Piha Vocalization Autocorrelation. Correlation outputs (top) Vocalization (bottom)

The mean of the respective vectors is subtracted from one-another, the dot product between the template and the unknown window is computed, and that value is divided by the product of the computed standard deviations and the number of elements in the feature vector.

To make sure that the number of correlation outputs is equal to the number of time bins of the STFT of the unknown signal, we perform zero-padding. Furthermore, when the correlation operation is performed and windows for the 2-dimensional STFT image must be selected, only windows where the frequencies perfectly align are selected, leaving only horizontal motion along the time axis. We like to think of this like "rollercoaster" cross-correlation. An example of ZNCC using the Xeno-canto template from Figure 2.2 on another "unknown" Xeno-canto clip can be seen in Figure 2.3. In the figure it is clear that peaks occur over birdsong calls that are similar to the template. We refer to the sequential correlation outputs between the template and the windows of an unknown signal as local score arrays.

2.3.3 Clustering Local Scores for Timestamps

In order to identify potential audio clips for humans to verify, we must convert the local score arrays into timestamps. To convert the local score arrays into timestamps, we set a threshold

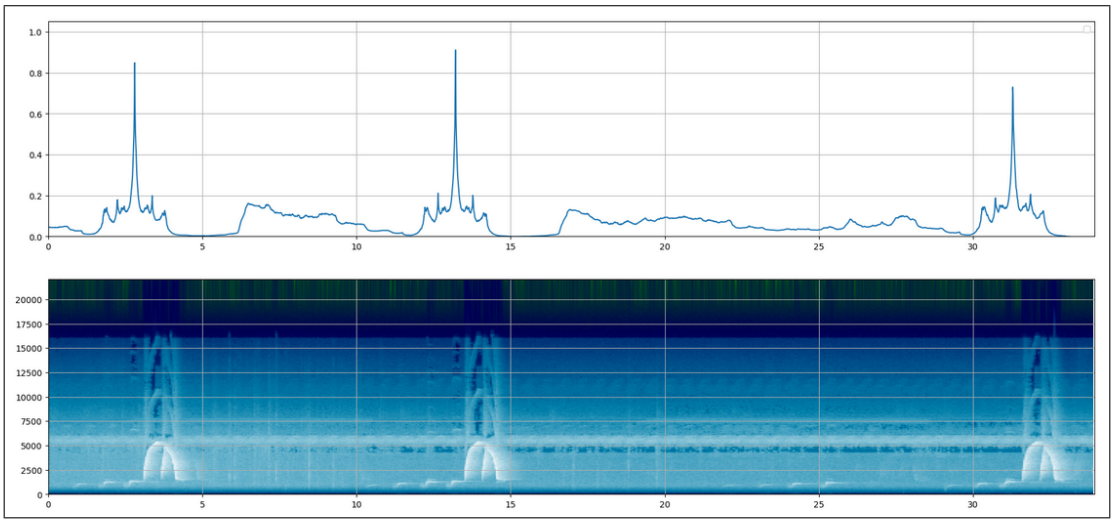


Figure 2.3. Screaming Piha Vocalization Cross Correlation. Correlation outputs (top) Unknown Signal (bottom)

with respect to correlation and place a window approximately equal to the length of the original template centered around the array elements greater than or equal to the threshold. We do not set the window size to be equal to the length of the original template to remain consistent across different templates from the same species we will run experiments on. All overlapping windows are converted into a single timestamp. This means that the timestamps can have variable lengths. Our team refers to this method as the "Steinberg Technique" in honor of the team member that first implemented this clustering method. An example of collecting timestamps using this method can be seen in Figure 2.4.

2.3.4 Statistical Learning Ensemble

After template matching has been performed, the statistical learning model ensemble will be used to filter out false positives. As noted in Figure 2.1, this template matching workflow involves two rounds of labeling. The first stage involves randomly sampling from the ZNCC audio timestamps to train an ensemble that is then used to help decide the remaining audio timestamps to label. For example, if we decide to label 100 template matching outputs, 20 will be used to train the ensemble, while the remaining 80 will be selected based on the ensemble

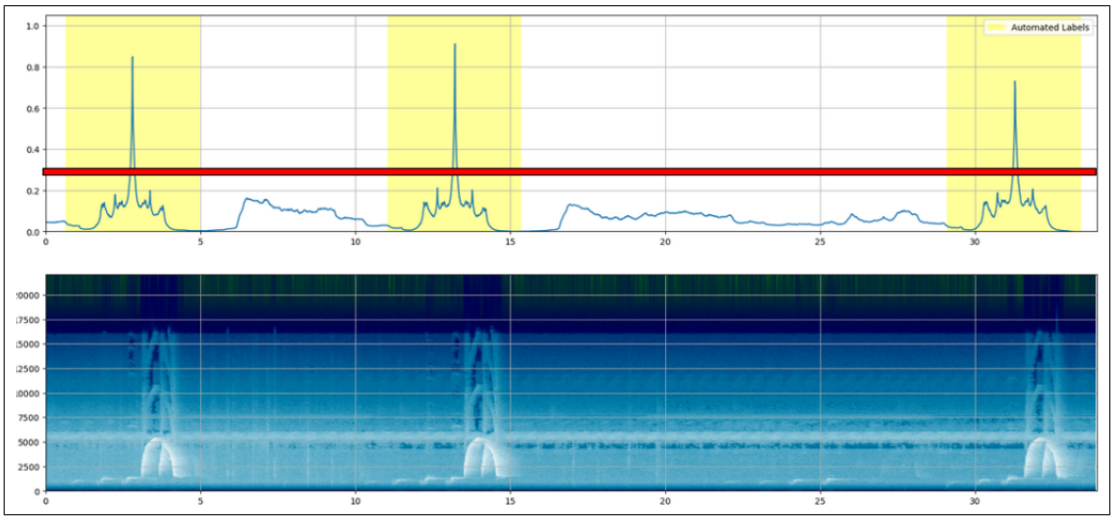


Figure 2.4. Screaming Piha ZNCC Timestamp (yellow) Creation based on Threshold (red)

vote.

In data-scarce environments, statistical learning models are often more appropriate than deep learning models because they typically require fewer labeled examples to achieve good performance [34]. Statistical learning methods, such as decision trees, support vector machines, and logistic regression, are well-suited for this scenario and can provide robust results with limited data.

Ensembling, the process of combining multiple models to improve overall performance, is a common technique in machine learning. By training several different models and combining their predictions, ensembling can reduce variance and bias, leading to more accurate and reliable results [19]. In this context, all models in the ensemble are trained for binary classification, where the presence of the relevant bird species vocalization is labeled as 1, and the absence (false positives) is labeled as 0.

The remainder of this section will provide brief overviews of each of the statistical learning models that make up the ensemble.

2.3.4.1 Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models used for classification and regression tasks, which work by finding the hyperplane that best separates data points of different classes. The linear SVM uses a straight hyperplane, while the radial (RBF kernel) SVM uses a curved hyperplane to handle non-linearly separable data. We use both kinds of boundary selection methods as separate models in the ensemble.

2.3.4.2 Logistic Regression

Logistic regression is a statistical method that can used for binary classification. It models the relationship between one or more independent variables (features) and a binary dependent variable (outcome). The method first computes a linear combination of the input features by calculating the weighted sum of the features plus a bias term. This linear combination is then passed through the logistic function, also known as the sigmoid function. The logistic function maps any real-valued number into a value between 0 and 1, representing the probability of the outcome. The core idea is to find the best-fitting model to describe the relationship between the dependent variable and independent variables by estimating the parameters (weights) that maximize the likelihood of observing the given data.

2.3.4.3 K-nearest Neighbor

K-nearest neighbor (kNN) is a method by which known points are plotted into some input space. Some unknown vectors can then be plotted and have their distance measured from all of the labeled data. An unknown vector can then be classified based on its nearest neighbor, or an arbitrary number of k nearest neighbors can be used to aggregate a class prediction.

2.3.4.4 Random Forest

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes of the individual trees.

2.3.5 Ensemble Prediction Aggregation and Sampling

After all of the models in the ensemble make a presence/absence class prediction across the unlabeled template-matched audio, we give each segment a number equal to the sum of the species present predictions. In this case where we have 5 statistical learning models, that means each clip gets a value [0,5]. We can then rank the remaining unlabeled audio in descending order based on this ensemble vote. We then select however many clips we are willing to label. To reconnect with the prior example, if we agreed to label 100 clips, given that we have already labeled 20 to train the ensemble, we would select the remaining 80 clips that received the highest ensemble vote.

2.4 Alternative Verification Methods

As is apparent from the long flow chart 2.1, the approach we are taking to make template matching verification more efficient is a very complicated process. In such situations, it is important to perform alternative experiments with simpler approaches. To which, we will run two alternative methods that will follow the template matching step and replace the approach to feeding audio segments for humans to verify. Both methods involve a single labeling stage where the agreed upon number of n clips to be labeled are labeled all at once. As opposed to our two stage approach where we labeled 0.2*n to train the ensemble and selected the remaining 0.8*n based on the vote of said ensemble. The first alternative approach will involve randomly sampling n clips from the template-matched audio. The second alternative approach will assign the max correlation to each template-matched audio segment that occurred over its duration. The template-matched audio segments are then placed in descending order and the top n segments are selected to be verified by humans. This is done with the hope that template-matched audio with higher correlation to the template are more likely to be true positives.

2.5 Experiments

As the chapter title and methodologies suggest, our proposed template matching experiments were developed with the intent of tackling passive recordings, what we called Type 3 recordings in Chapter 1 section 1.1.3. With that being said, we will still test the methodology on purposeful recordings with and without weak labels, what we called Type 1 and Type 2 recordings. We do this to be thorough in our comparisons on how to tackle each of these kinds of recordings.

2.5.1 Data

Our experiments are focused around finding training data for the Screaming Piha, Bright-rumped Attila, and the Common Poorwill that are all described in Chapter 1 section 1.2. We derive two Type 3 passive datasets. The first includes Audiomoth recordings from a single device in our Madre de Dios, Peru deployment. The second includes all of the audio clips from the Scripps Coastal Reserve deployment as it is generally a less noisy environment and less prone to false positives. For our purposeful audio recordings from Xeno-canto we create a Type 1 weakly labeled dataset by selecting clips known to have the species of interest. We then simulate a Type 2 purposeful dataset without weak labels by adding in clips with species known to exist in the respective regions that have weak labels that are not the species of interest.

2.5.2 Template Matching Setup

All of the Madre de Dios (MDD), Scripps Coastal Reserve (SCR), and Xeno-canto Templates for the Screaming Piha, Bright-rumped Attila, and Common Poorwill are shown in Figure 2.6. Table 2.1 show all of the parameters used for template matching. The confidence threshold of the Steinberg technique was selected by some experimental runs where a species template was run against a clip known to contain matches and evaluating the local score arrays. The sampling rate for all of the audio in each template matching run was selected based on

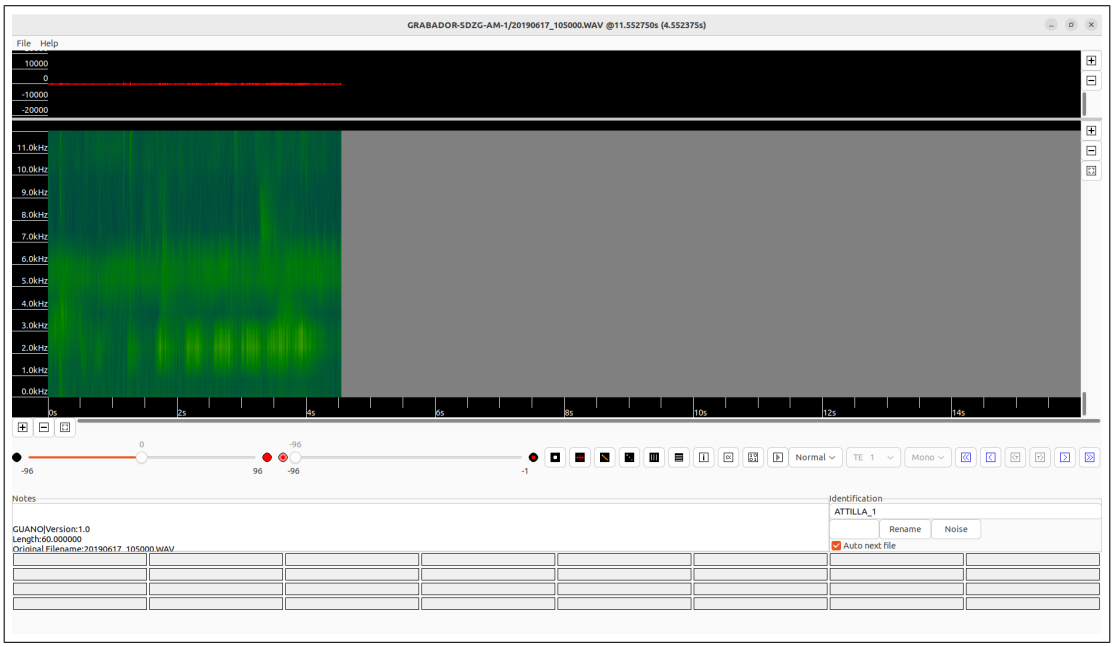
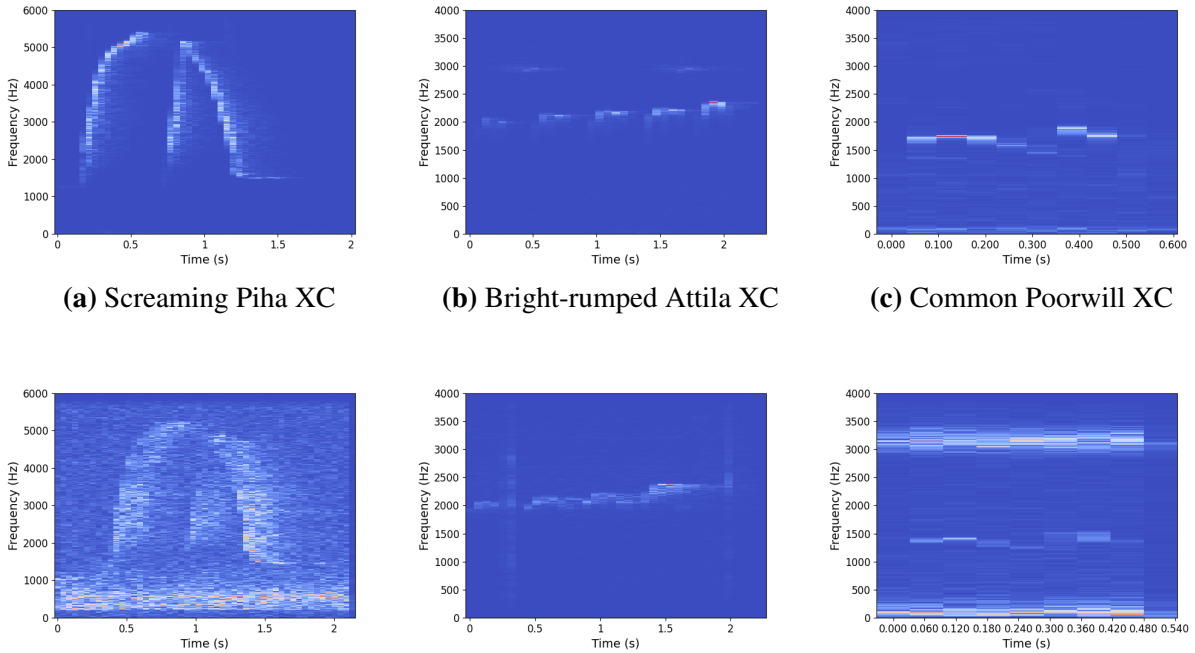


Figure 2.5. Example Kaleidoscope Lite Interface for Template Matching Verification

the highest frequency of each relevant call. We could have chosen a lower frequency for the Common Poorwill experiments. The frequency passband of the Butterworth bandpass filter, defined by the ”lowcut” lower bound frequency and ”highcut” upper bound frequency were selected based on lower and upper frequency bounds of each species template. The window for the Steinberg technique was selected to be approximately equal to the length of the templates used. We acknowledge that the Screaming Piha window size is as bit of an outlier with respect to the length of the template used. This error likely stems from an original intent to use templates that include both the main ”pee-haw” calls alongside the preamble ”whoo” calls that was shifted away from when we used only the main calls for templates.

All of the template matching audio outputs were verified using Wildlife Acoustic’s Kaleidoscope Lite software. This was chosen as it makes it easy to quickly listen to audio segments with an accompanying spectrogram visual and label for presence/absence of the template species. An example of this software interface is shown in Figure 2.5.



(a) Screaming Piha XC **(b)** Bright-rumped Attila XC **(c)** Common Poorwill XC

(d) Screaming Piha MDD **(e)** Bright-rumped Attila MDD **(f)** Common Poorwill Scripps

Figure 2.6. Purposeful and Passive Recording Templates

2.5.3 Statistical Learning Ensemble Setup

As mentioned in the Methodology, the statistical learning ensemble is the critical step with respect to the two labeling stages. With each of the statistical learning models, there are parameters that control the behavior of the model. In this work, we simply work with the default values used by Scipy [70]. The one challenge we must contend with as a consequence of our clustering algorithm outputting variable-length time stamps is how to make the input vectors during training and inference have a uniform length. We solve this by setting the length to be slightly longer than the window size selected for each species. Table 2.1 shows these window lengths. For the audio segments shorter than the input space, we simply repeat the clip until it is greater than the desired input space and concatenate. For the audio segments that are greater than the desired input space, we simply cut off any excess length.

2.5.4 Evaluation

As mentioned in prior sections, all of the outputs from our template matching runs have been verified by humans. Since we perform template matching on audio data that is not completely strongly labeled, we can only identify true positives (TPs) and false positives (FPs). That is to say that since we did not exhaustively strongly label the entire dataset, we do not know what was missed by template matching, we cannot count the number of false negatives (FNs). With that, we can calculate the performance of our template matching via precision -

$$\text{precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs}}$$

By labeling all of the template matching outputs across all of the Type 1, Type 2, and Type 3 recordings across all species, we are able to perform simulated runs where we limit the number of n clips to be verified. This enables us to compare the percentage of the template-matched audio segments that have been verified to the percentage of the overall number of true positives that have been successfully extracted from the unlabeled pool. With that, we can draw an ideal upper bound where every clip that is labeled is a true positive until only the false positives remain. Similarly, we can generate a lower bound where every clip that is sampled is a false positive until only true positives remain. For example in a situation where there are 1000 template-matched datapoints with 600 true positives, if we simulate a labeling run with 500 datapoints, the upper bound point would be (0.5, 0.833) while the lower bound point would be (0.50, 0.167). From this, we can then see where our three approaches to template matching verification, using ensemble learning, random sampling, and high-correlation sampling, fall between these upper and lower bounds.

To evaluate the performance of each method, we first need to quantify their effectiveness. This is achieved by calculating the area under the curve (AUC), which provides a single scalar value summarizing the overall performance of a template matching verification strategy across all

simulated runs. The AUC represents the method's accuracy and reliability, offering an intuitive measure of its effectiveness. To further refine this evaluation, we compare the AUC of each model to the ideal AUC, representing perfect performance, by taking a ratio. This AUC ratio quantifies how close each methodology is to the ideal, providing a clear and concise measure of performance. The AUC is calculated using Simpson's Rule [11].

2.6 Results

Figure 2.7 shows a histogram of the correlation scores of each template matching run. Here we can see that for the Xeno-canto recordings with weak labels, there are generally many more true positives than false positives. This should not be a surprise that if we take a template of a species of interest and perform template matching on purposeful audio recordings that generally have more focal sounds of interest and we know the species of interest exists (Type 1), then we are likely to have a very successful template matching run with minimal false positives. In the next row where template matching is performed on purposeful recordings without weak labels (Type 2), we can see that more false positives are introduced. This makes sense as more audio clips were included where the species of interest did not exist. For the Screaming Piha and Bright-rumped Attila, we can generally see that the true positives are skewed towards higher correlations. However, for the Common Poorwill we can see an unexpected result that the false positives have generally higher correlation values than the true positives. This is likely due to the Common Poorwill having a call that is fairly short in length and occupies a small frequency bandwidth making it prone to correlate with other bird vocalizations. In the final row of template matching runs on passive recordings (Type 3) we can see that the Bright-rumped Attila had TPs and FPs that were mostly separable based on correlation. However, the Screaming Piha and the Common Poorwill template matching runs yield messy histograms where TPs and FPs are intermingled.

Figure 2.8 shows a breakdown of all of the simulated labeling runs. In the first row of

Type 1 recordings we can see that there is minimal space between the upper and lower bounds. This means that almost any strategy to separate TPs from FPs will work. This should make sense from the confidence histogram row of Type 1 recordings that had very few FPs relative to TPs. In the next row of Type 2 recordings we can see that for the Screaming Piha and Bright-rumped Attila, simply labeling the template-matched audio with the highest correlation performs the closest to ideal. This makes sense as in their respective confidence histograms, we could see that the TPs and FPs were largely separable with TPs being generally higher than FPs. However, we can see how assuming that the TPs will have higher correlation can pose problems with the Common Poorwill as that approach performs worse than simply randomly selecting template-matched audio to label. Finally, in the third row of template matching runs on Type 3 recordings, we can see that the ensemble approach to template matching verification performs as well as or better than sampling based on correlation.

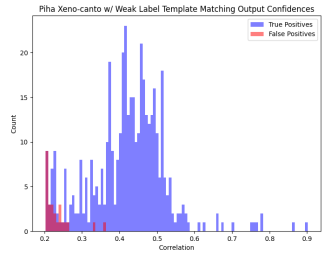
Table 2.2 shows a more quantitative breakdown of the full template matching runs as well as the AUC ratios from the simulated template matching runs from Figure 2.8. In that we can see how the precision of the Type 1 recordings are very high as there were very few positives. The Type 2 and Type 3 recordings generally have lackluster precision falling below 60% with the exception of the Type 3 Bright-rumped Attila.

While Table 2.2 does include the AUC ratio results, for the sake of discussing said results, we will look at Figure 2.9 that shows the AUC Ratios in a side-by-side bar chart with respect to the relevant datasets. In this figure in the Type 1 recordings in the first row, we can see that all of the techniques achieve close to ideal results which was also observed with the minimal difference between the upper and lower bounds of Figure 2.8. In the second row of Type 2 recording AUC Ratio results we can once again see how sampling template-matched audio based on high correlation achieves the results closest to the ideal upper bound for the Screaming Piha and Bright-rumped Attila. However, this approach does not generalize across all species as it yields worse results compared to naively random sampling in the case of the Common Poorwill.

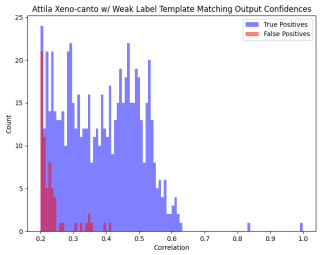
From these results, the most important thesis contribution is that the ensemble approach

Table 2.1. Parameters Used for Template Matching and Ensemble Learning

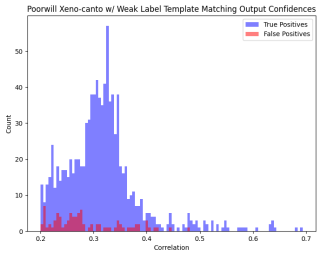
Dataset	**Corr. Threshold**	**Steinberg Window Length (s)**	**Bandpass Lowcut (Hz)**	**Bandpass Highcut (Hz)**	**Sample Rate (Hz)**	**Ensemble Window Length (s)**
Attila_XC_w_Weak	0.2	2	1000	3500	8000	3
Piha_XC_w_Weak	0.2	4.2	850	5600	12000	4.5
Poorwill_XC_w_Weak	0.2	0.6	1000	2000	8000	1
Attila_XC_wo_Weak	0.2	2	1000	3500	8000	3
Piha_XC_wo_Weak	0.2	4.2	850	5600	12000	4.5
Poorwill_XC_wo_Weak	0.2	0.6	1000	2000	8000	1
Attila_AM	0.18	2	1000	3500	8000	3
Piha_AM	0.18	4.2	850	5600	12000	4.5
Poorwill_AM	0.2	0.6	1000	2000	8000	1



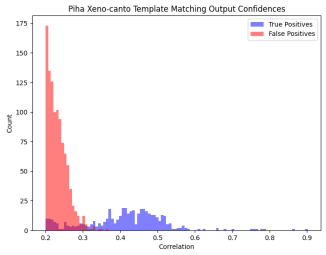
(a) Screaming Piha XC w/ Weak Label



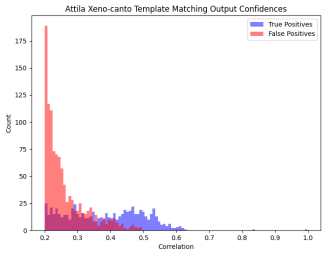
(b) Bright-rumped Attila XC w/ Weak Label



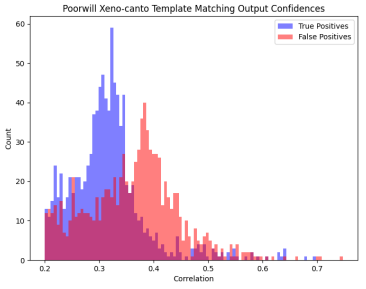
(c) Common Poorwill XC w/ Weak Label



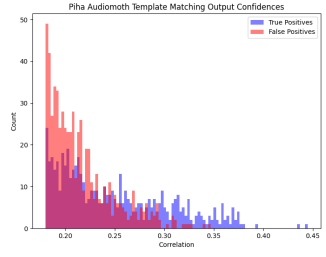
(d) Screaming Piha XC



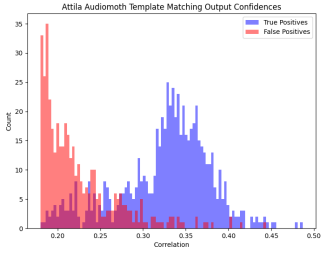
(e) Bright-rumped Attila XC



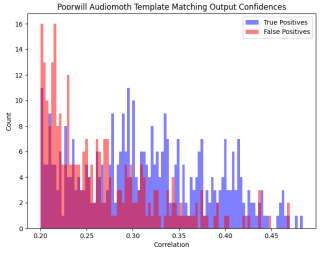
(f) Common Poorwill XC



(g) Screaming Piha Audiomoth

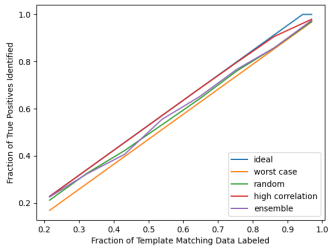


(h) Bright-rumped Attila Audiomoth

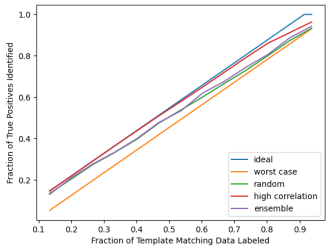


(i) Common Poorwill Audiomoth

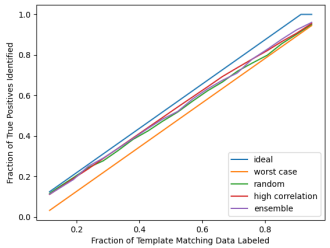
Figure 2.7. Template Matching Confidence Histograms



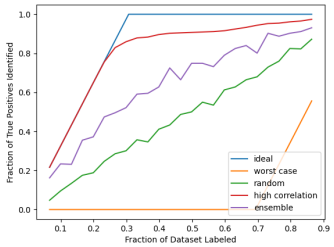
(a) Screaming Piha XC w/ Weak Label



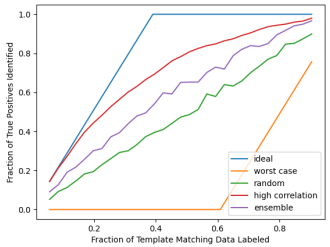
(b) Bright-rumped Attila XC w/ Weak Label



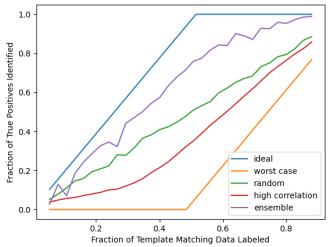
(c) Common Poorwill XC w/ Weak Label



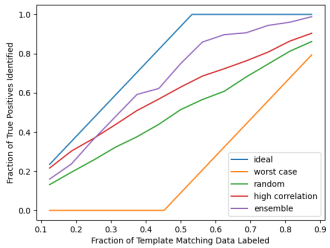
(d) Screaming Piha XC



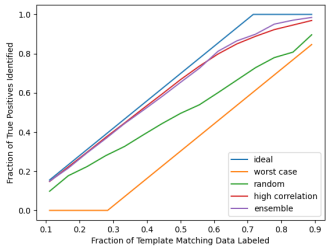
(e) Bright-rumped Attila XC



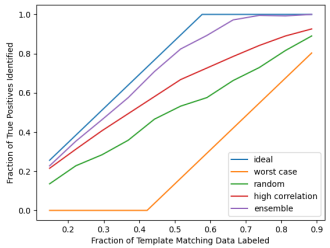
(f) Common Poorwill XC



(g) Screaming Piha Audiomoth



(h) Bright-rumped Attila Audiomoth

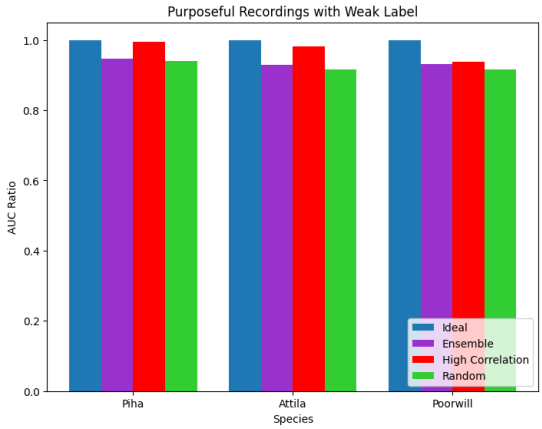


(i) Common Poorwill Audiomoth

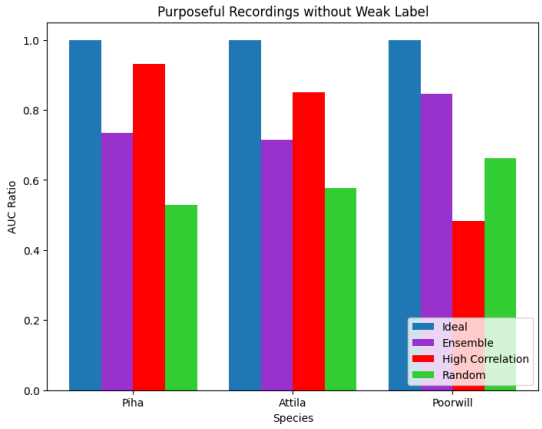
Figure 2.8. Simulated Template Matching Verification Runs

Table 2.2. Template Matching Verification Summary

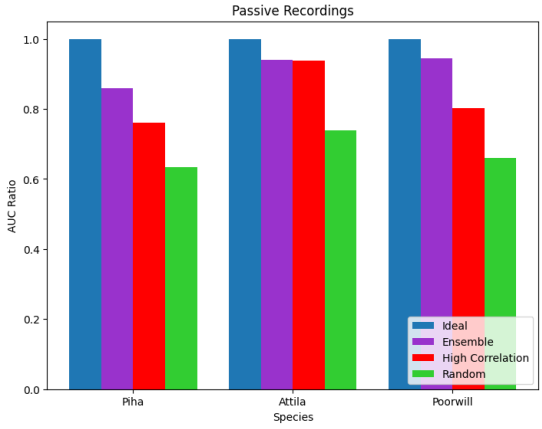
Species	Dataset	Clip Count	Template Matching Output Count	TP Count	Prec.	Random AUC Ratio from Ideal	High Correlation AUC Ratio from Ideal	Ensemble AUC Ratio from Ideal
Piha	XC w/ Weak	146	464	438	0.944	0.941	0.994	0.947
Piha	XC	528	1505	462	0.307	0.528	0.932	0.735
Piha	MDD	2877	800	426	0.541	0.633	0.760	0.858
Attila	XC w/ Weak	123	747	683	0.914	0.916	0.982	0.929
Attila	XC	362	1769	691	0.391	0.576	0.850	0.714
Attila	MDD	2877	899	644	0.716	0.738	0.938	0.941
Poorwill	XC w/ Weak	168	964	1053	0.916	0.917	0.938	0.932
Poorwill	XC	358	1874	966	0.516	0.633	0.483	0.845
Poorwill	Scripps	20114	671	391	0.578	0.660	0.802	0.945



(a) Purposeful Recordings w Weak Labels



(b) Purposeful Recordings wo Weak Labels



(c) Passive Recordings wo Weak Labels

Figure 2.9. AUC Ratio Side-by-side Bar Charts

to template matching verification consistently performs as well as, or better than sampling based on high-correlation in passive recordings. This is best highlighted in Figure 2.8 where in the third row, the ensemble curve is consistently the closest to the ideal upper bound. This result is further supported by the passive recordings side-by-side bar chart in 2.9 where the AUC ratio for the ensemble from the aforementioned curves is the greatest, though only marginally so in the case of the Bright-rumped Attila.

2.7 Chapter Acknowledgements

Chapter 2 uses data from an Audiomoth deployment in La Jolla, California. The thesis author led this Audiomoth deployment that was funded by Engineers for Exploration under the leadership of Curt Schurgers and Ryan Kastner.

Chapter 2 uses data from an Audiomoth deployment in Madre de Dios Peru. The audio data collected from Peru was kindly provided to us by our San Diego Zoo Wildlife Alliance collaborators Mathias Tobler and Ian Ingram.

Chapter 2 describes the use of an algorithm to cluster temporal predictions into timestamps. This algorithm has yet to appear in a publication. It was developed by Gabriel Steinberg.

Chapter 3

Extracting Training Data from Purposeful Recordings

Many researchers and hobbyists collect audio recordings in natural environments with the intent of collecting bird vocalizations. When these datasets are provided to the scientific community, they often only specify a single species that can be heard. They do not indicate when these vocalizations occur. That is to say, that scientists interested in using this audio data to train deep learning models that can differentiate between different species vocalizations often cannot use the data directly from the source. The only information that is actually known about the audio is that at least one vocalization of the species of interest has occurred. Given that many deep learning models require inputs of some fixed length, in the case of audio, some fixed time, not knowing when the vocalizations occur can become an issue for larger audio recordings. Some of those issues include -

1. Audio Sparsity
    (a) Example - a minute long clip contains only 3s of the vocalization of interest.
    (b) Risk - training and consequently making inferences on environmental background noise as the class of interest.
2. False Positive Sounds
    (a) Example - a human voice noting the time and date of the recording.

(b) Example - a species of bird other than the one labeled makes a sound.

(c) Risk - neural network model struggles to differentiate between vocalizations of interest.

3. Species Vocalization Variance

(a) Example - a dataset is labeled for a species of interest but only contains mating calls.

(b) Example - a training dataset contains only foreground vocalizations when a study area contains background calls.

(c) Risk - neural network model struggles to generalize for the species of interest.

In the data acquisition section of the paper that describes the current state-of-the-art deep learning model for acoustic species identification, BirdNET [36], they describes the challenge of - "training on weakly labeled samples of varying lengths, the biggest challenge is to extract segments of audio recordings that contain the target signal." This section, alongside this paper by Morfi [50], led to our adoption of the nomenclature that separate "weak" labels, or what we call Type 1 recordings, where the presence of at least one vocalization of a species exists within a purposeful audio clip, compared to "strong" labels that provide the exact timestamps of where said species can be heard within a clip. They go on to reference work done by Sprengel [62] for the 2016 BirdCLEF competition that details an automated approach to foreground-background separation to isolate focal vocalizations within a weakly labeled audio clip that was used in the training of BirdNET. We will refer to such methods that attempt to automate selection of relevant timestamps from weakly labeled audio as weak-to-strong (WTS) pipelines.

In the remaining sections of the chapter we will cover WTS methods developed by other researchers. For those that we build on, we will cover how our work varies from said work. Our methodologies section will motivate each of our approaches while also providing more details on each WTS pipeline and how they are applied to Type 1 recordings. We will then cover experiments that compare the training data generated by each WTS pipeline to those created

by humans. Finally, we will cover experiments related to the training and evaluation of deep learning models for acoustic bird species classification on the training data generated by the WTS pipelines. The models will be evaluated on Type 1, Type 2, and Type 3 recordings.

3.1 Related Works

In this section we will describe work done by fellow researchers with respect to building WTS pipelines. Some of these works we build directly on top of and so we will mention those extensions, but go into more detail in the methodologies portion of the chapter.

3.1.1 Foreground-background Sound Separation

Given that someone has purposefully shared an audio clip that they recorded where they identified a species of interest, it seems logical that the vocalization of interest would be the most prominent sound in the audio clip. So, if a researchers wants to extract said vocalization from all the other audio, it would make sense to come up with an algorithm that simply separates foreground audio from background audio.

The de-facto standard WTS pipeline for training deep learning models for bird acoustic species identification from Sprengel [62] uses such an approach. In this paper they describe a process, of creating a binary mask over a spectrogram using power thresholding, followed by some binary morphology operations, and converting the mask into a temporal indicator vector. We consider this the de-facto standard due to its usage in BirdNET [36]. We reverse-engineer this WTS pipeline and compare it to other similar methodologies.

Once they have a temporal indicator vector, they concatenate all the regions of interest together, and then they split that new audio segment into uniform segments for training. Furthermore, they take the regions considered to be in the background, that fall below a power threshold, follow the same concatenation process, and additively combine with the foreground audio to create a new sample. This process is shown in Figure 3.1

Furthermore, the authors do explicitly state that they use the foreground-background

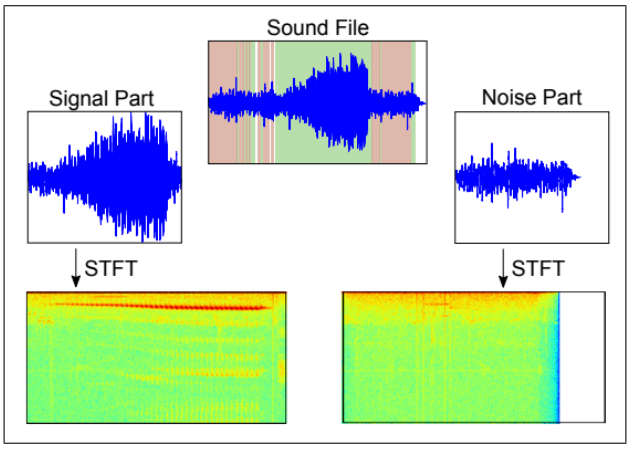


Figure 3.1. Example of Foreground-background (green-red) concatenation from Sprengel et al, 2016

separation technique as a pre-processing step during inference time after the neural network has been trained 1. While the BirdNET paper did cite the Sprengel paper as their weak to strong labeling pipeline, we don't know for certain that they used the exact same hyperparameters that control the algorithm for training. Furthermore, we do not know if they use the foreground-background separation as a pre-processing step at inference time

When it comes to the use of the their foreground-background separation technique for pre-processing, it brings with it the challenge of mapping those back to specific time segments. For example, say we have a concatenated 5 second audio segment that was extracted from a minute long audio segment. Assume that this 5 second segment is created from a 3 second segment in the first half of the audio clip and a 2 second segment from the second half. If our model then predicts the presence of some species, do we then have to store some metadata about where in the clip those segments came from? Furthermore, we then have to verify where in the original audio clip the species was heard. Now, say we ignore that challenge and instead only use their concatenation process for training where they are concatenating together discontinuities in the audio data. When they go to inference on all the audio of a test set, the underlying data has some fundamental differences where one was exposed to a pre-processing step while the other has not. These challenges of course can be worked around, it just seems that they lead to

some complications that can be avoided for the sake of simplicity.

Due to these challenges, in our application of this foreground-background separation technique, we take a different approach to converting the temporal indicator vector into training data. We will show how these changes make it easier to compare to a human ground truth as well. Furthermore, we will not treat it as a pre-processing step when we evaluate model inference performance.

3.1.2 Binary Birdsong Sound Event Detection

Given the challenge of extracting vocalizations of a species of interest from an audio clip where the presence of said species is known to exist, a logical choice would be to train a general model that identifies sounds from the relevant species taxon. For our work that would come in the form of a sound event detection model trained to generally identify bird vocalizations.

The closest work in binary bird sound event detection related to our goals in this thesis comes from Morfi [50]. In that paper they describe a Convolutional Neural Network and Recurrent Neural Network (CNN-RNN) hybrid architecture that attempts to identify when bird vocalizations occur in an audio clip. They train, a WHEN network that identifies temporal regions of interest and a WHO network that discriminates between bird species. They go into detail about different ways to efficiently train the networks together. That is, they maximize the value of a data scarce environment by sharing the weights of the feature extraction convolutional layers of both networks and compare against independently training the WHEN and WHO networks.

Our work makes use of an open-source implementation of the WHEN network that was trained on the DCASE 2018 [39] (Warblr and Freefield1010) datasets. This open source implementation is referred to as Microfaune [31]. Our work varies from that of Morfi by using the pre-trained Microfaune model as an approach to the weak to strong label challenge to create training data for a multi-species classifier.

Similar work that leverages a CNN-RNN architecture come from Cohen and Nicholson

with TweetyNet [13]. They build out a birdsong sound event detection model that performs well in classifying what kind of Bengalese finch and canary syllables occur and when they do so in audio. Their study used training and test data meant for behavioral studies, meaning that it was collected indoors.

We will run experiments with the TweetyNet architecture using weights trained across a larger variety of species for general binary bird sound event detection. The training data comes from outdoor recordings [49]. We will go into more depth on how Microfaune and Tweetynet are used in this study in this chapter's methodologies.

3.1.3 Template Matching

The goal of this chapter is to extract the exact timestamps of vocalizations from a species of interest in an audio clip where the presence of that species is confirmed. Assuming we have access to at least one fine-grain example template of a sound from that species, it is a logical choice to use this template to extract similar sounds from the weakly labeled clip.

As described in section Chapter 2, template matching can be used to identify species vocalizations within an audio clip. Therefore, we consider the previously discussed template matching literature in section 2.1 as related work to this challenge as well.

The application of template matching in Chapter 2 involved focusing massive passively recorded Type 3 datasets down to a more focused set of points that correlate with a species of interest. In this chapter, the method will be applied to weakly labeled purposeful recordings to extract sounds similar to a species of interest which are then compared directly to human labels on the same audio data. Furthermore, we will use the template-matched audio data to train neural networks and compare the performance across other WTS pipelines as well as human labeled data.

3.2 Methodology

A general overview of our methodology can be seen in Figure 3.2. The general idea is that we are given an audio clip that is known to contain a bird vocalization of interest. From there, we pass this audio clip through a Weak-to-strong pipeline that identifies where the relevant bird vocalizations occur. From there, we must normalize those timestamps into input feature vectors of uniform length to train neural networks.

3.2.1 Weak to Strong Labeling Pipelines

In this section we will list out all of the WTS pipelines we use to generate estimates of a strongly labeled training set from weakly labeled audio clips. We will emphasize a high-level motivation behind each approach as well as acknowledge their drawbacks. We will list out the technical details for each approach and fill in smaller algorithm parameters and training details in the Experiments section 3.3.

3.2.1.1 Naive

When labeled purposeful audio clips with weak labels are collected from online resources with the intent of training deep learning models, the simplest approach is to assume that all of the clip contains the weak label species of interest. As most deep learning models require some fixed input length, the naive approach is to simply break the audio data into chunks equal to the length of the fixed input, and use everything as training data. For shorter segments, such as the last chunk of an audio clip or audio clips less than the required fixed input length, one must decide to either repeat the clip until it is the required length, or to simply skip over such clips. We opt for the latter approach. The danger in this approach is that by accepting all the audio data, you open up the risk of including false positives that are not the species of interest such as background noise and sounds of different species.

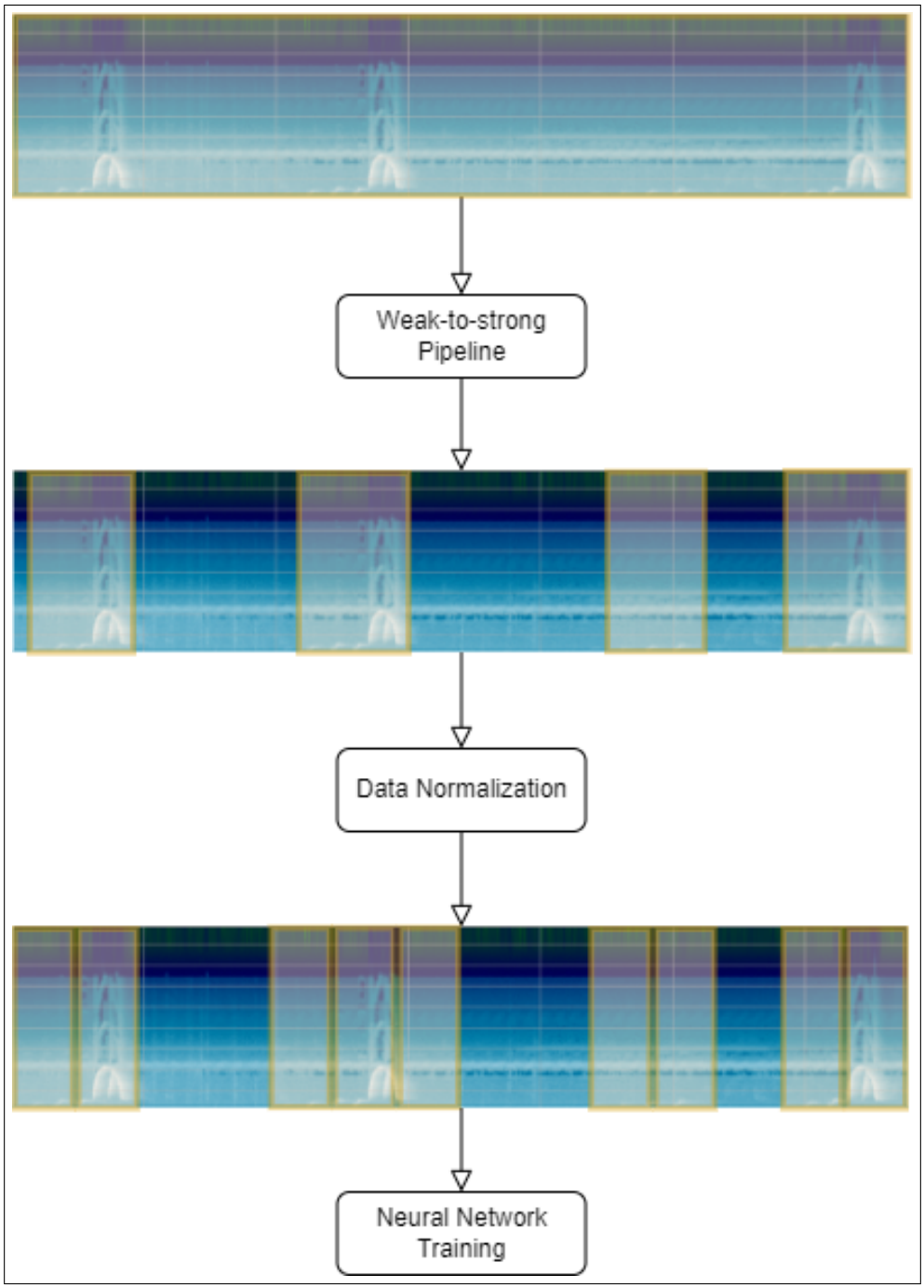


Figure 3.2. Weak-to-strong (WTS) Pipeline Flowchart

3.2.1.2 Foreground-background Separation

As mentioned in section 3.1.1, we have reverse-engineered the foreground-background separation method used in [62]. Their method is summarized in Algorithm 1. An example of this method applied to a weakly labeled Xeno-canto audio clip can be seen in Figure 1.

Algorithm 1. Sprengel et al. 2016 Foreground-background separation.

Compute the STFT with a Hanning window of length 512, with 75% overlap
Take the absolute value of the STFT
Normalize the STFT to [0, 1] by dividing by the maximum value
Compute the median for every row and column of the normalized STFT
Select a constant multiplier b that defines the power threshold.
Construct binary mask:
for each pixel in the normalized STFT **do**
  if pixel $>$ b $\times$ row_median AND pixel $>$ b $\times$ column_median **then**
    Set pixel to 1
  else
    Set pixel to 0
  end if
end for
Select integer constant c that defines the dimensions of your binary morphology kernels.
Apply binary morphology opening operation with a c x c square kernel of 1's:
Apply erosion (nested 2D AND operation) with kernel
Apply dilation (nested 2D OR operation) with kernel
Convert "opened" binary mask to time indicator vector (binary local-score array):
Compute the sum of each column (creating a vector of sums)
for each value in the vector **do**
  if value $\geq$ 1 **then**
    Set to 1
  end if
end for
Apply dilation two times on the time indicator vector with a c x 1 kernel.
*Note that this is similar to a convolution operation where the kernel is flipped, so, a 1 x c kernel will end up actually being applied to the indicator vector

The challenge with any foreground-background WTS pipeline is that it relies on the assumption that the weak label is the dominant sound within the audio clip. For the bird vocalizations we are interested in, this assumption can be problematic. Loud anthropogenic sounds, such as human voices, footsteps, vehicles, or recorder shuffling, can interfere. Additionally, sounds

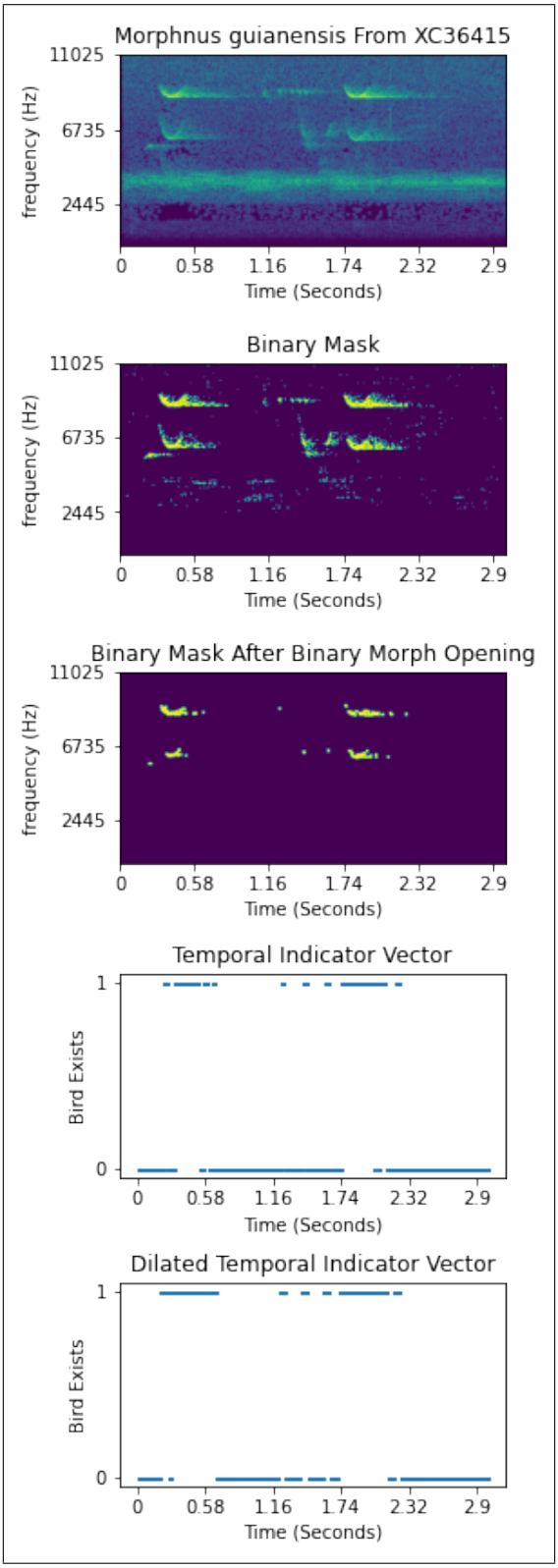


Figure 3.3. Sprengel et al. 2016 Foreground-background Separation Demonstration

from other fauna, including incorrect taxa like frogs and insects or calls from incorrect bird species, can exist in the foreground of the audio. Without a human in the loop to double-check the training data, it is possible that these potential sources of false positives end up in the training set with incorrect labels, which can yield a model that struggles to classify the species of interest.

3.2.1.3 Binary Bird Sound Event Detection

To automate the process of converting weak labels in purposeful datasets into strong label timestamps of the labeled bird species, we explore binary bird sound event detection. The idea behind this is that given a certain bird species vocalization occurs at least once, if we can separate bird calls from non-bird calls within an audio clip, we can use those separations for training data on said species.

We work with two bird audio Sound Event Detection CNN-RNN hybrid models. The first, we refer to as Microfaune, is an open source recreation of the WHEN network described in [50] [31]. The second is the TweetyNet architecture developed by [13]. Both neural networks at a high-level take the same approach of having a feature extraction block of convolutional layers that identify regions of interest within spectrograms which is then followed by a block that learns temporal correlations using bidirectional long-short-term-memory (LSTM) layers.

Two of the main differences between the architectures is that TweetyNet was designed to distinguish between types of vocalization syllables, meaning that each prediction in the local score array output is multi-dimensional. We train the model in a purely binary manner. Furthermore, Tweetynet performs a local score array post-processing step where all strings of local scores that are shorter than a user-defined minimum duration are removed. For the strings above said threshold, a majority vote within all of the predictions in the segment is performed so each segment only falls under one class.

There are two fundamental challenges with using a binary bird sound event detection model to automate the WTS process. The first challenge is that even if you have a perfect model that can perfectly identify bird presence/absence, the weakly labeled audio can contain species

vocalizations that are not the relevant species. The second challenge is rooted in the inherent complexity of training a deep learning model that generalizes across all species. That is to say, that the breadth of species audio used to train the sound event detection model may fail to generate model weights that can be applied to a species in a weakly labeled dataset.

3.2.1.4 Template Matching

In a situation where a researcher interested in collecting training data for a species has access to at least one vocalization example, what we call a template, and has access to weakly labeled recordings, the logical step is to leverage said template to find similar sounds. As previously discussed, this is a process called template matching.

We use the same zero-normalized cross correlation (ZNCC) approach to template matching described in Chapter 2 section 2.2.3 followed by the Steinberg technique to acquire timestamps.

The main concern with this approach as a WTS pipeline is that a vocalization template will correspond to a certain call of a species. This is a problem when most bird species have several kinds of calls. An example we discussed earlier in section 1.3.4 is how the Screaming Piha has a main ”pee-haw” call as well as the preamble ”whoo” call. If the goal is to train a deep learning model that can generally classify a certain species, an implicit assumption is that it can identify a wide variety of calls said species makes. Hence, training data derived from this WTS pipeline does not account for that variety as it will be skewed towards sounds similar to the template vocalization used.

3.2.2 Training Data Length Normalization

As shown in the methodology flowchart in Figure 3.2, assuming a WTS pipeline has processed an audio clip, one must come up with some scheme to create uniform length training data so that it can be used to train a neural network. This is due to the fact that the image classification CNN neural networks that we use require fixed input lengths [42].

Many of the WTS pipelines output local score arrays that contain values over [0,1]. In the case of a binary sound event detection model, these correspond to the model prediction of the presence of a bird vocalization in a given time segment. In the case of ZNCC template matching, these correspond to how much a given time segment correlates with the template. In the foreground-background separation technique, these local score arrays are binary, or what Sprengel [62] refers to as a temporal indicator vector. For the other techniques, we have described how timestamps are acquired by various means such as the Steinberg technique described in Chapter 2 section 2.3.3. These timestamps, that represent the presence of the species of interest within the audio, can similarly be treated as a temporal indicator vector where 1 represents the presence of the weak label species and 0 represents the absence of the weak label species.

Once a temporal indicator has been created, an audio clip can be ”chunked” into uniform length time segments based on the desired length of the input feature vector for the neural network. From there, if a chunk contains any local scores of 1, that indicate the presence of the species of interest, that chunk can be saved as training data. Everything else is thrown out. In a case where the desired chunk length does not evenly fit into an audio clip, we simply ignore the excess. For example, say we desire 5 second long chunks from an 11 second long audio clip, we will ignore the last second. We do this because the alternatives such as repeating the remaining 1 second audio segment 4 times or padding 4s worth of 0s seem unrealistic for the data the model will ultimately make inferences on.

3.2.3 Multi-species Classification Deep Learning Model

Assuming that training data has been collected we now must consider which model to use. When it comes to selecting a deep learning model for a task, one must decide what kind of architecture to use for a given application. We choose to work with Convolutional Neural Networks due to their compelling performance on image classification, remote sensing research [66]. Furthermore, training a robust deep learning model from scratch can be expensive and time consuming. To address this problem, foundational models that have been trained on large image

classification datasets can be used and trained in process referred to as transfer learning [56]. EfficientNet models meet both of these requirements.

The term EfficientNet encompasses a family of CNNs designed for image classification tasks. They are known for their high performance and efficiency. Developed by Google, EfficientNet uses a novel scaling method called ”compound scaling” that uniformly scales all dimensions of depth, width, and resolution using a set of fixed scaling coefficients [65]. We take advantage of the Hugging Face timm API [72] that encapsulates several state of the art image classification models. From timm, we use Efficientnet-b0, the smallest of the Efficientnet architectures. We choose the smallest architecture to better emulate a scenario that is accessible to a wider range of researchers interested in training models for acoustic species identification.

3.2.4 Weak Label Aggregation

Now that we have selected a model to our satisfaction, we must figure out how to evaluate the models we train. Given that this thesis addresses the challenges of training data availability, we also face similar challenges with the availability of testing data. Specifically, we lack robust strongly labeled data of all of the relevant species across all Type 1, Type 2, and Type 3 recordings.

To address this, we will use results from Chapter 2, where we identified species vocalizations using template matching. During the template matching verification process, we confirmed the presence of at least one species vocalization of several species across both passive and purposeful datasets. These confirmations of species presence can be converted into weak labels across an audio clip. For instance, we performed template matching using a Screaming Piha vocalization across thousands of Type 3 recordings. From that we got a few hundred regions that were labeled as Screaming Piha or not Screaming Piha. In the case of a segment of an audio clip that did contain a Screaming Piha, we can now accept Screaming Piha as a weak label across the whole audio clip.

With these weakly labeled datasets, when an EfficientNet model performs inference on

these clips, it will generate multiple predictions depending on the input size. These predictions will be aggregated to determine the weak label for each test clip. For example, say we have a 15 second audio clip and a model that makes inferences on 5 second audio chunks. Furthermore, assume that this model has been trained on 3 classes. Assuming we inference on the 15 second audio clip without overlap, we will end up with a 3 x 3 matrix where the rows represent each 5 second chunk and the columns represent the predictions over [0,1] of the 3 classes. The question now is how to convert this 3 x 3 matrix into a single prediction of the weak label of the clip. The first thing we will do is take the mean with respect to the clips which outputs a 1 x 3 row vector that represents the average prediction of each class. From there we accept the class with the greatest mean prediction as the weak label for the clip. There are many approaches that are similar to this in the field of multiple instance learning [10]. This is the approach we went with after seeing some successes with it in some initial proof of concept experiments involving training deep learning models with WTS pipelines that are not included in this thesis.

3.3 Experiments

In this section we will cover the more minute details of each of the methodology flowchart 3.2 such as the specific datasets used, algorithm parameters selected, model training, and how we evaluate both the WTS pipelines compared to human strong labels and the final multi-species classifier performance.

3.3.1 Human Strongly Labeled Data

In order to compare these WTS pipelines in their capacity to strongly label a dataset, we must have some human-labeled ground truth for which to compare. To create a human-labeled training data baseline, Xeno-canto clips with weak labels for Screaming Piha and Bright-rumped Attila species were strongly labeled. To strongly label the audio data, Pyrenote was used due to its server-side nature, making it easier to access across devices, and its simple strong labeling interface. An example of this interface is seen in Figure 3.4.

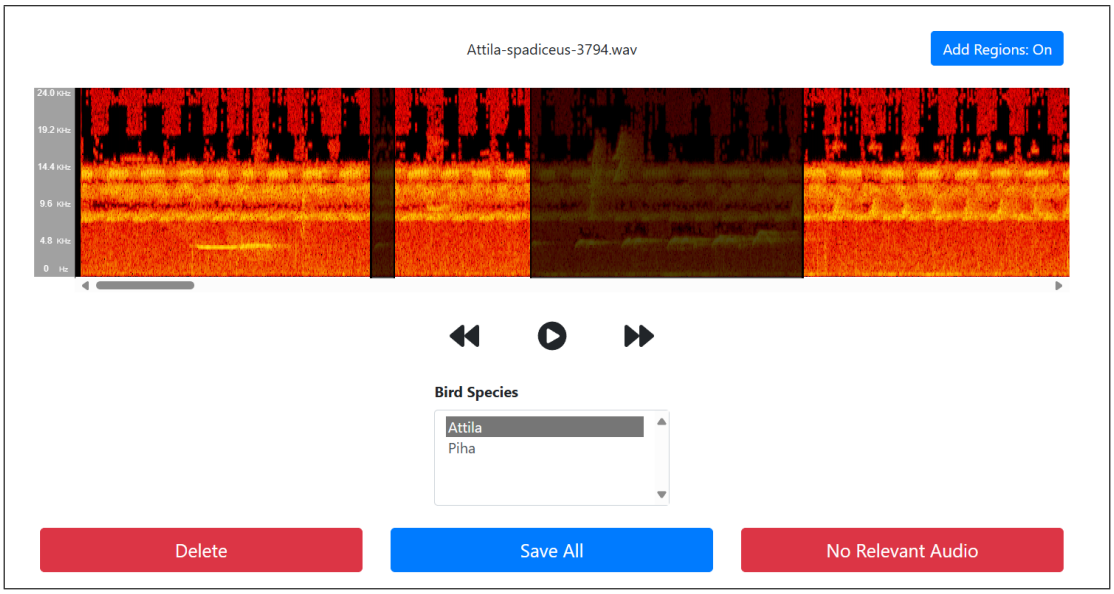


Figure 3.4. Pyrenote Interface for Strong Labeling on a Bright-rumped Attila Xeno-canto clip

3.3.2 Generating Temporal Indicator Vectors with each WTS Pipelines

In this section we go into more detail on the intricacies of each of the WTS pipelines for the sake of reproducibility. We will specifically describe how we derive an indicator vector from each pipeline on the Madre de Dios, Peru bird species dataset before the data length normalization process.

3.3.2.1 Naive

As described in the methodology, this WTS pipeline involves assuming that all of the audio within the weakly labeled Xeno-canto recordings contain the species of interest. This means that the temporal indicator vector will be an array of 1's signifying the presence of the weak label species across the entire audio clip.

3.3.2.2 Foreground-background Separation

We use the algorithm described in 1 where we set the power threshold constant "b" to 3 and we set the kernel dimension constant to "4". These parameters are used as they were the parameters selected from the algorithm authors in Sprengel [62].

3.3.2.3 Microfaune

The open-source version of the model described in [50], called Microfaune, from this repository [31] has two key differences from the original paper. First, is that Microfaune was trained on the DCASE 2018 competition datasets "freefield1010" and "warblr10k". In the original Morfi paper, their WHEN network was trained on NIPS4b [49]. The freefield1010 dataset contains 7690 field recordings from around the world and the warblr10k contains 8000 citizen science smartphone audio recordings from the United Kingdom. These datasets were turned into 10s long spectrogram feature vectors. The second major change that they made from the Morfi paper was that they used regular a regular Cross Entropy loss function, whereas Morfi used a specialized loss function derived from the field of multiple-instance learning. We will re-iterate that we did not train the version of Microfaune used in these experiments, instead opting to use the pre-trained weights available in the project's Github repository.

The only addition we made relates to converting the local score array inference output to a temporal indicator vector. For that, we set two thresholds on the local score arrays. We set a static lower-bound threshold of 0.12 as well as a dynamic threshold set to be 3.2 times greater than the median of the local score array of the clip being processed. We set these two thresholds since the training data species are completely different what Microfaune was trained with and the data we are evaluating. This often leads to lower confidences, but relative peaks still occurring where bird vocalizations appear. Hence, the use of the dynamic threshold. We use the lower static threshold so as to not include too many predictions the model is not confident on. We set the Steinberg window size to 2 seconds.

3.3.2.4 TweetyNet

Since the original TweetyNet architecture was trained for multi-class sound event detection, we decided to train the model from scratch using NIPS4B [49]. Since NIPS4B is a human strongly labeled dataset, we can safely train a bird-present class 1 and bird-absent class 0. The input spectrograms were derived from audio data fixed to 44.1 kilohertz with a window length

2048 with 50% overlap. The STFT was then converted into the mel scale with 72 mel bins. The model was trained with a learning rate of 0.005 for 500 epochs using cross-entropy loss.

To create a temporal indicator vector we take the argmax class prediction between bird presence and absence for each local score prediction. Any contiguous segment of 1's that is shorter than 2 seconds in length is thrown out. This is generally the same methodology for prediction post-processing described in [13].

3.3.2.5 Template Matching

We used the same Xeno-canto templates used for the Screaming Piha and Bright-rumped Attila purposeful recording experiments in Chapter 2. Said templates can be seen in Figure 2.6. Furthermore, the Steinberg technique and bandpass filtering parameters can be seen in Table 2.1.

3.3.3 Evaluating Weak to Strong Label Pipelines

As mentioned earlier, since we strongly labeled the audio data the WTS pipelines were applied to, we can quantitatively compare each WTS pipeline to the human ground truth. First, we must select the training data length we desire. We choose to go with 3 seconds, echoing the feature vector length of BirdNET [36]. Since the human annotations are of variable length, we apply the same 3s chunking normalization process that precedes all the WTS pipeline. With everything normalized we can then look at the intersecting training datapoints as true positives (TPs), WTS exclusive training points as false positives (FPs), and human exclusive training points as false negatives (FNs). This allows us to calculate the metrics -

$$\text{precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs}}; \text{recall} = \frac{\text{TPs}}{\text{TPs} + \text{FNs}}; \text{f1} = \frac{\text{2*precision*recall}}{\text{precision} + \text{recall}}$$

3.3.4 Efficientnet Training

In order to compare the WTS pipelines in their ability to replicate human labels, we also want to compare the performance of the same multi-class classification model trained to

differentiate between Screaming Piha and Bright-rumped Attila calls across all five of the WTS pipelines as well as the human ground truth.

All models were trained using transfer learning from the Efficientnet-b0 model that has been pre-trained on the ImageNet dataset [18]. All models were trained for 100 epochs, using a binary cross-entropy with logits loss function at a learning rate of 0.001 with a 90-10 train-validation split. Mel spectrograms were generated using a hanning window of length 1024 with 50% overlap and 192 mel bins. Furthermore, for reproducibility sake, all of the random seeds are set to 0 so all the training starts with the same model weights.

3.3.5 Efficientnet Evaluation

As previously mentioned in the methodology, since we do not have access to strongly labeled test sets, we must instead make do with weakly labeled data we have available. This leaves us with the task of aggregating neural network predictions across a clip to try and predict the weak label.

Since our EfficientNet models were all trained on 3s segments. When we perform inference we must aggregate all of the 3s predictions across a clip into a prediction on the weak label. We take the average across all the 3s predictions across each class. This leaves us with a mean prediction for each class. We can then take the argmax across these mean predictions across a clip and use the greatest mean class as a prediction of the weak label. We compare these predicted weak labels to the actual weak label. This is far from an ideal way to evaluate neural network performance on a bird classification task, but it is what we went with given our aforementioned testing data limitations.

Since we did not train the EfficientNet model on any classes other than Screaming Piha and Bright-rumped Attila, we assume any predictions that fall below a threshold of 0.95 are an ”other” class. Such a high threshold was selected as we found the predictions in general to be very skewed right over the prediction range [0,1]. With these predictions and the other class, we can define a model correctly prediction a weak label as a TP, a model incorrectly predicting a

weak label as a FP, a model calling a Screaming Piha or Bright-rumped Attila clip other as a FN, and a model correctly predicting a clip labeled as other as a true negative (TN). Furthermore, this means that for the case of the weakly labeled, purposeful dataset, there are not any true ”other” clips to identify as we only include clips where the relevant bird species are known. This allows us to calculate the aforementioned precision, recall, and f1 metrics.

3.4 Results

We summarize the results of each WTS pipelines compared to a human-labeled ground truth in Table 3.1. In it we can see the number of training datapoints collected, as well as the precision, recall, and f1. We can see that TweetyNet achieves the highest precision, the naive approach has the highest recall, which makes sense since there are no false negatives when you collect everything, and the foreground-background separation technique achieves the highest f1, though it is inconsequentially better than TweetyNet. Since we consider false positives (FPs) to be undesirable for a training set, as they can make the model struggle to identify the true positive (TP) species of interest, we consider Tweetynet's higher precision as an improvement over the foreground-background separation given the insignificant difference in f1. This is because it is important to minimize false positives without going to the extreme of having no training data available. Completely eliminating false positives can result in a lack of sufficient positive examples, which is crucial for the model to learn and generalize effectively.

Table 3.1. Weak to Strong Labeling Pipeline Training Estimation Summary

WTS	Count	Precision	Recall	F1
Human	3034	1.0	1.0	1.0
Naive	3753	0.80	1.0	0.89
FGBG	3540	0.84	0.98	0.91
Microfaune	2219	0.84	0.69	0.76
Tweetynet	2997	0.91	0.90	0.90
Template Matching	2884	0.85	0.85	0.85

Next, we look at the results of deep learning models trained on the WTS pipeline training

datasets on the task of weak label prediction for bird species.

Table 3.2 summarizes all of the deep learning models trained on the 5 WTS pipelines as well as the human-labeled ground truth and evaluated on test sets of all three Types of recordings. For each training dataset, across each recording Type, we see the number of number of clips in each test set as well as the number of TPs, FPs, FNs, and TNs. It also includes the precision, recall, and f1 performance metrics. In this we can see how the Type 1 recordings do not have TNs, which is due to the fact that by definition of Type 1 recordings, we do not include data with incorrect weak labels. This leads to less FPs as the only source of FPs come from calling a Screaming Piha clip a Bright-rumped Attila and vice versa. This small number of FPs leads to the very high precision for Type 1 recordings. For the Type 2 test set we can see that after the human baseline the naively trained model achieves the highest precision. This is also true for the Type 3 test set. These high naive precisions are a surprise given that this is the model trained with the greatest number of FPs as seen in Figure 3.1. We do not have a strong hypothesis as to why this might be occurring and do not wish to dive into too much speculation given this less than ideal approach to evaluating these models. Across all of the experiments, the recall performance was rather poor, with the highest being 0.7192 for the Microfaune applied to Type 1 recordings. This low recall is due to the high confidence threshold we set at 0.95 which will give many clips a prediction of ”other.”

In this case, we view f1 as the most important metric for evaluation as it balances precision and recall. To which, we show the side-by-side f1 bar charts in Figure 3.5. In the first subfigure of Type 1 tests, we can see that there is little difference in the f1 performance across the different WTS pipelines with the exception of template matching. The poor performance of template matching is likely due to the test set containing a wide variety of Screaming Piha and Bright-rumped Attila calls whereas the template matching train set is skewed towards a single call type for each species. In the second subfigure related to Type 2 recordings, there is little discernible difference between the f1 performance across the networks. Looking at the last subfigure we can see that TweetyNet achieves the highest f1 score which seems to align with

our claim that it does the best job of estimating a training dataset. However, since that training dataset estimate claim is in comparison to the human labels that were used to train a model as well, we cannot claim that TweetyNet both is a better estimate and has the best performance if the human-labeled training set led to worse model performance.

To summarize the chapter results, we do have evidence to suggest that the binary bird sound event detection approach to automating the weak to strong label process does the best job of estimating a human ground truth. This is given by the high precision and f1 performances of TweetyNet in Table 3.1. With that being said, given that the human ground truth performs worse than what we claim is its best automated estimate, we do not have evidence to claim that it leads to better deep learning performance on the task of acoustic species identification. This is best highlighted in Figure 3.5 where we can see that TweetyNet achieves the highest f1 performance, including against a model trained on human-labeled recordings we claim is a ground truth. Any hypothesis why TweetyNet outperformed a human-labeled ground truth would be mostly speculative. Instead, we will attribute the odd results to a rather lackluster evaluation setup of trying to aggregate weak labels.

3.5 Chapter Acknowledgements

Chapter 3 contains screenshots from our team's manual audio labeling software, Pyrenote. Sean Perry was the lead developer of the software. The thesis author helped delegate resources and advised the development.

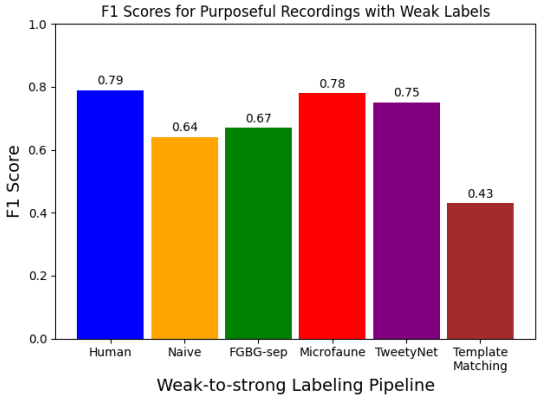
Chapter 2 uses data from an Audiomoth deployment in Madre de Dios Peru. The audio data collected from Peru was kindly provided to us by our San Diego Zoo Wildlife Alliance collaborators Mathias Tobler and Ian Ingram.

Chapter 3 describes the use of an algorithm to cluster temporal predictions into timestamps. This algorithm has yet to appear in a publication. It was developed by Gabriel Steinberg.

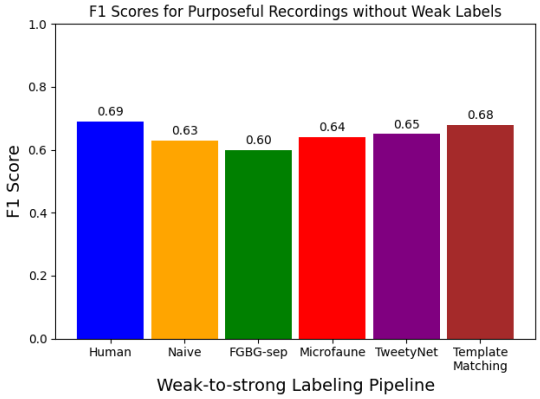
Chapter 3 describes a bird sound event detection neural network called TweetyNet. The

Table 3.2. Performance Metrics of EfficientNet Models on the Weak Label Prediction Task

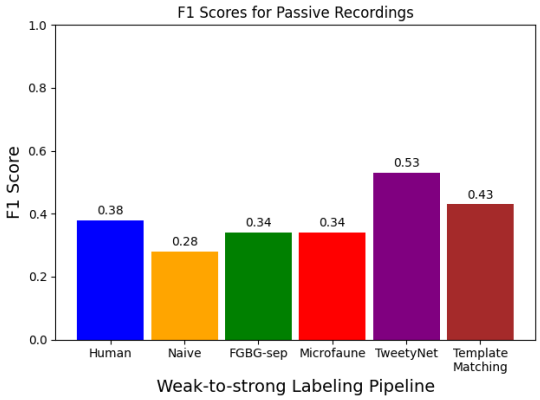
	TPs	**FPs**	**FNs**	**TNs**	**Total**	**Precision**	**Recall**	**F1**
Weak_Naive	182	1	204	0	387	0.9945	0.4715	0.6397
Weak_FGBG	197	4	186	0	387	0.9801	0.5144	0.6747
Weak_Microfaune	274	6	107	0	387	0.9786	0.7192	0.8290
Weak_Tweetynet	249	7	131	0	387	0.9727	0.6553	0.7830
Weak_TM	231	10	146	0	387	0.9585	0.6127	0.7476
Weak_Human	253	3	131	0	387	0.9883	0.6589	0.7906
XC_Naive	191	25	196	339	751	0.8843	0.4935	0.6335
XC_FGBG	203	86	182	280	751	0.7024	0.5273	0.6024
XC_Microfaune	260	177	120	194	751	0.5950	0.6842	0.6365
XC_Tweetynet	252	144	126	229	751	0.6364	0.6667	0.6512
XC_TM	239	89	140	283	751	0.7287	0.6306	0.6761
XC_Human	252	91	132	276	751	0.7347	0.6563	0.6933
AM_Naive	49	4	250	395	698	0.9245	0.1639	0.2784
AM_FGBG	69	37	227	365	698	0.6509	0.2331	0.3433
AM_Microfaune	114	304	144	136	698	0.2727	0.4419	0.3373
AM_Tweetynet	154	136	138	270	698	0.5310	0.5274	0.5292
AM_TM	102	84	190	322	698	0.5484	0.3493	0.4268
AM_Human	82	57	213	346	698	0.5899	0.2780	0.3779



(a) Purposeful Recordings with Weak Labels



(b) Purposeful Recordings without Weak Labels



(c) Passive Recordings without Weak Labels

Figure 3.5. EfficientNet F1 Metrics Side-by-side Bar Charts

thesis author summarizes the training of the network as well as its application of estimating human training data. Our team has been using this model for a while, but it has not yet appeared in any publications. Mugen Blue was the team member that trained the TweetyNet weights used in this work.

Chapter 3 describes training the Efficientnet neural network architecture. To accomplish this, the thesis author used our team's multiclass training pipeline that has yet to be used in a publication. This pipeline was developed primarily by Sean Perry and Samantha Prestrelski.

Chapter 4

Conclusion

In this brief chapter we will reiterate the thesis contributions around acquiring bird species vocalization training data for deep learning models. It will also include a discussion on some of the drawbacks and what alterations could be made to the methodologies covered.

4.1 Contributions

Passive acoustic monitoring surveys tend to yield very large datasets. These surveys are often conducted to understand biodiversity health over an ecosystem. These datasets are often too large for manual human verification for acoustic species identification. To efficiently parse such datasets, researchers aim to use robust deep learning models that have demonstrated their efficacy in image classification. Given that audio data can be converted into spectrogram images, the challenge then becomes acquiring adequate training data for species vocalizations of interest.

We have categorized two fundamental kinds of datasets of which training data can be drawn from. We define passive datasets as those that are often set on periodic schedulers and are recording at times that are independent of the audio that is ultimately collected. We define purposeful datasets as those that have been recorded with a higher degree of dependence on the underlying audio. This can come in the form of an individual reactively recording upon hearing a sound of interest or audio that has been listened to and certain regions of interest have been made available to the scientific community. In general, for both training data acquisition and deep

learning evaluation, passive recordings are more challenging due to their lack of bias towards sounds of interest.

We categorize two different kinds of species labels for audio data. Weak labels are defined as those that confirm the existence of a species vocalization of interest within an audio clip. Strong labels specify when the species vocalization of interest occurs within the audio clip.

With these recording and label categories, we define two primary challenges. How should one go about acquiring training data for a species of interest directly from unlabeled passive recordings that maximizes the value of human verification? In the event that purposeful recordings that are weakly labeled for a species of interest are available, what is the most effective strategy to automate the strong labeling process to filter audio data that does not contain the species of interest?

To acquire training data from passive recordings, we leverage audio clips that contain a vocalization of a species of interest, referred to as a template. We use the template to find audio that is highly correlated and test out different schemes to select which template-matched audio is verified by humans. We find that breaking up the verification process into two stages is the most effective strategy on passive recordings. In the first stage a small random sample of the template matched audio is labeled and used to train an ensemble of statistical learning models to discriminate between true positives and false positives. These classifiers are then used to vote on which clips are most likely to contain true positives of the species vocalization of interest.

To acquire training data from purposeful recordings with weak labels, we test out several sound event detection techniques that attempt to estimate the underlying strong labels. These techniques include naively assuming all the audio has the species of interest, high signal-to-noise ratio foreground-background separation, binary bird classification, and template matching. When these methods are compared to training data acquired by humans strongly labeling weakly labeled audio, we find that a binary classifier known as TweetyNet performs the best. However, when EfficientNet models were trained on the human labels, as well as the automated labels on predicting weak labels of unknown audio, there was not strong evidence to suggest that the

TweetyNet training data performed the best.

4.2 Discussion

Overall, the work in this thesis related to automated weak to strong pipelines was motivated as our team saw what appeared to be a critical subsystem within a larger pipeline for training deep learning models for passive acoustic monitoring. Most papers brush off what is an estimation of a training dataset in a couple of sentences like an afterthought. To which, we have built out a framework to compare these training data estimates to a ground truth, independent of the actual final deep learning model performance.

A major potential drawback of some of these automated methods can be performance variance between classes. For instance, it is possible that the binary bird classifier can struggle with certain species. Furthermore, different species can have a very large variance in the number of weakly labeled clips that are available.

One idea we have considered but did not have the time to experiment with is cascading some of the WTS outputs together. For instance, what would happen if we take the intersection between the training set estimations of a binary classifier and a foreground-background separation algorithm? Furthermore, since the onset of this project, some transformer based sound event detection models have been developed [24] that could be compared to TweetyNet and Microfaune.

Furthermore, we hope that our expansion of the work of [4] can help bring the notion of making template matching more efficient to a wider audience. Particularly as the first steps when a passive acoustic monitoring dataset is collected can be quite a daunting task.

The largest drawback on our work in template matching verification is that it is ultimately dependent on the performance of the initial template matching step. That is to say, for species that have very short calls in frequency bands where other noises are common, these methods likely will not perform well.

For the two stage ensemble verification, we did some initial experiments where we tried

different sampling techniques to decide which clips were used to train the ensemble. The first involved a 50-50 split between the highest and lowest correlation template-matched clips. The second involved a 33-33-34 split between the highest correlation, lowest correlation, and random template-matched clips. We did not expand on these tests as they did not seem to affect the results in our initial experiments.

Bibliography

[1] Edith B Allen, Christopher McDonald, and Bridget E Hilbig. Long-term prospects for restoration of coastal sage scrub: Invasive species, nitrogen deposition, and novel ecosystems1. In *Restoration Workshop*, page 1, 2019.

[2] Alexandre Antonelli, Alexander Zizka, Fernanda Antunes Carvalho, Ruud Scharn, Christine D Bacon, Daniele Silvestro, and Fabien L Condamine. Amazonia is the primary source of neotropical biodiversity. *Proceedings of the National Academy of Sciences*, 115(23):6034–6039, 2018.

[3] Marcelo Araya-Salas and Grace Smith-Vidaurre. warbler: an r package to streamline analysis of animal acoustic signals. *Methods in Ecology and Evolution*, 8(2):184–191, 2017.

[4] Cathleen M Balantic and Therese M Donovan. Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring. *Bioacoustics*, 29(3):296–321, 2020.

[5] Kathryn Bowler, Pavel Castka, and Michaela Balzarova. Understanding firms' approaches to voluntary certification: Evidence from multiple case studies in fsc certification. *Journal of Business Ethics*, 145:441–456, 2017.

[6] Peter A Bowler. Ecological restoration of coastal sage scrub and its potential role in habitat conservation plans. *Environmental Management*, 26:S85–S96, 2000.

[7] Roel JW Brienen, Oliver L Phillips, Ted R Feldpausch, Emanuel Gloor, Tim R Baker, Jon Lloyd, Gabriela Lopez-Gonzalez, Abel Monteagudo-Mendoza, Yadvinder Malhi, and Simon L Lewis. Long-term decline of the amazon carbon sink. *Nature*, 519(7543):344–348, 2015.

[8] Keith S Brown Jr. Conservation of neotropical environments: insects as indicators. *The conservation of insects and their habitats*, 349:404, 1991.

[9] Roberto Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[10] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[11] Kenneth V Cartwright. Simpson's rule cumulative integration with ms excel and irregularly-spaced data. *Journal of Mathematical Sciences & Mathematics Education*, 12(2):1–9, 2017.

[12] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[13] Yarden Cohen, David Aaron Nicholson, Alexa Sanchioni, Emily K Mallaber, Viktoriya Skidanova, and Timothy J Gardner. Automated annotation of birdsong with a neural network that segments spectrograms. *Elife*, 11:e63853, 2022.

[14] Juan Colonna, Tanel Peet, Carlos Abreu Ferreira, Alípio M Jorge, Elsa Ferreira Gomes, and João Gama. Automatic classification of anuran sounds using convolutional neural networks. In *Proceedings of the ninth international c* conference on computer science & software engineering*, pages 73–78, 2016.

[15] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[16] Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767 – 1781, 2011.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[19] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[20] Johan T Du Toit. Considerations of scale in biodiversity conservation. *Animal Conservation*, 13(3):229–236, 2010.

[21] Sara Fraixedas, Andreas Lindén, Markus Piha, Mar Cabeza, Richard Gregory, and Aleksi Lehikoinen. A state-of-the-art review on birds as indicators of biodiversity: Advances, challenges, and future directions. *Ecological Indicators*, 118:106728, 2020.

[22] Christine M Gabriele, Janet L Neilson, Janice M Straley, C Scott Baker, Jennifer A Cedarleaf, and James F Saracco. Natural history, population dynamics, and habitat use of humpback whales over 30 years on an alaska feeding ground. *Ecosphere*, 8(1):e01641, 2017.

[23] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time fourier transform. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983.

[24] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, and Yonghui Wu. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[25] Fred Harris. *Let's Assume the System Is Synchronized*, pages 311–325. Springer Netherlands, Dordrecht, 2011.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Nicole E Heller and Erika S Zavaleta. Biodiversity management in the face of climate change: a review of 22 years of recommendations. *Biological conservation*, 142(1):14–32, 2009.

[28] Andrew P Hill, Peter Prince, Evelyn Piña Covarrubias, C Patrick Doncaster, Jake L Snaddon, and Alex Rogers. Audiomoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods in Ecology and Evolution*, 9(5):1199–1211, 2018.

[29] Muhammad Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint arXiv:1706.07156*, 2017.

[30] Denise Jäckel, Kim G Mortega, Ulrike Sturm, Ulrich Brockmeyer, Omid Khorramshahi, and Silke L Voigt-Heucke. Opportunities and limitations: A comparative analysis of citizen science and expert recordings for bioacoustic research. *Plos one*, 16(6):e0253763, 2021.

[31] Hadrien Jean. Microfaune, 2020. Accessed: 2021-05-10.

[32] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

[33] Gareth Jones, David S Jacobs, Thomas H Kunz, Michael R Willig, and Paul A Racey. Carpe noctem: the importance of bats as bioindicators. *Endangered species research*, 8(1-2):93–115, 2009.

[34] Shahab Eddin Jozdani, Brian Alan Johnson, and Dongmei Chen. Comparing deep neural networks, ensemble classifiers, and support vector machine algorithms for object-based urban land use/land cover classification. *Remote Sensing*, 11(14):1713, 2019.

[35] Kantapon Kaewtip, Abeer Alwan, Colm O'Reilly, and Charles E Taylor. A robust automatic birdsong phrase classification: A template-based approach. *The Journal of the Acoustical Society of America*, 140(5):3691–3701, 2016.

[36] Stefan Kahl, Connor M Wood, Maximilian Eibl, and Holger Klinck. Birdnet: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, 61:101236, 2021.

[37] Jonathan Katz, Sasha D Hafner, and Therese Donovan. Tools for automated acoustic monitoring within the r package monitor. *Bioacoustics*, 25(2):197–210, 2016.

[38] Derek Keeping and Rick Pelletier. Animal density and track counts: understanding the nature of observations based on animal movements. *PLoS one*, 9(5):e96598, 2014.

[39] Qiuqiang Kong, Turab Iqbal, Yong Xu, Wenwu Wang, and Mark D Plumbley. Dcase 2018 challenge surrey cross-task convolutional neural network baseline. *arXiv preprint arXiv:1808.00773*, 2018.

[40] Sam Lapp, Tessa Rhinehart, Louis Freeland-Haynes, Jatin Khilnani, Alexandra Syunkova, and Justin Kitzes. Opensoundscape: An open-source bioacoustics analysis package for python. *Methods in Ecology and Evolution*, 14(9):2321–2328, 2023.

[41] Sam Lapp, Nickolus Stahlman, and Justin Kitzes. A quantitative evaluation of the performance of the low-cost audiomoth acoustic recording unit. *Sensors*, 23(11):5254, 2023.

[42] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[43] Daniel W Leger and D James Mountjoy. Geographic variation in song of the bright-rumped attila (tyrannidae: Attila spadiceus): implications for species status. *The Auk*, 120(1):69–74, 2003.

[44] Jiazhi Liang. Image classification based on resnet. In *Journal of Physics: Conference Series*, volume 1634, page 012110. IOP Publishing, 2020.

[45] Adria Lopez-Baucells, Ricardo Rocha, Paulo Bobrowiec, Enrico Bernard, Jorge Palmeirim, and Christoph Meyer. *Field Guide to Amazonian Bats*. 09 2016.

[46] Melanie L Low, Mairelys Naranjo, and Jayne E Yack. Survival sounds in insects: diversity, function, and evolution. *Frontiers in Ecology and Evolution*, 9:641740, 2021.

[47] Peter R Marler and Hans Slabbekoorn. *Nature's music: the science of birdsong*. Elsevier, 2004.

[48] Ryan Mcgeady, Robert M. Runya, James S. G. Dooley, John A. Howe, Clive J. Fox, Andrew J. Wheeler, Gerard Summers, Alexander Callaway, Suzanne Beck, Louise S. Brown, Gerard Dooly, and Chris Mcgonigle. A review of new and existing non-extractive techniques for monitoring marine protected areas. *Frontiers in Marine Science*, 10, July 2023. © 2023 McGeady, Runya, Dooley, Howe, Fox, Wheeler, Summers, Callaway, Beck, Brown, Dooly and McGonigle. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY).

[49] Veronica Morfi, Yves Bas, Hanna Pamuła, Hervé Glotin, and Dan Stowell. Nips4bplus: a richly annotated birdsong audio dataset. *PeerJ Computer Science*, 5:e223, 2019.

[50] Veronica Morfi and Dan Stowell. Deep learning for audio event detection and tagging on low-resource datasets. *Applied Sciences*, 8(8):1397, 2018.

[51] Harini Nagendra. Using remote sensing to assess biodiversity. *International journal of remote sensing*, 22(12):2377–2400, 2001.

[52] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2:1–21, 2015.

[53] Erwin Nemeth. Measuring the sound pressure level of the song of the screaming piha lipaugus vociferans: One of the loudest birds in the world? *Bioacoustics*, 14(3):225–228, 2004.

[54] Tom H Oliver and Mike D Morecroft. Interactions between climate change and land use change on biodiversity: attribution problems, risks, and opportunities. *Wiley Interdisciplinary Reviews: Climate Change*, 5(3):317–335, 2014.

[55] Paula Fernanda Pinheiro Ribeiro Paiva, Maria de Lourdes Pinheiro Ruivo, Orleno Marques da Silva Júnior, Maria de Nazaré Martins Maciel, Thais Gleice Martins Braga, Milena Marília Nogueira de Andrade, Paulo Cerqueira dos Santos Junior, Eduardo Saraiva da Rocha, Tatiana Pará Monteiro de Freitas, and Tabilla Verena da Silva Leite. Deforestation in protect areas in the amazon: a threat to biodiversity. *Biodiversity and Conservation*, 29:19–38, 2020.

[56] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[57] Sean Perry, Vaibhav Tiwari, Nishant Balaji, Erika Joun, Jacob Ayers, Mathias Tobler, Ian Ingram, Ryan Kastner, and Curt Schurgers. Pyrenote: a web-based, manual annotation tool for passive acoustic monitoring. In *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 633–638. IEEE, 2021.

[58] Rajeev Rajan and A Noumida. Multi-label bird species classification using transfer learning. In *2021 International Conference on Communication, Control and Information Sciences (ICCISc)*, volume 1, pages 1–5. IEEE, 2021.

[59] Stefan Schneider, Graham W Taylor, and Stefan Kremer. Deep learning object detection methods for ecological camera trap data. In *2018 15th Conference on computer and robot vision (CRV)*, pages 321–328. IEEE, 2018.

[60] Ivan W Selesnick and C Sidney Burrus. Generalized digital butterworth filter design. *IEEE Transactions on signal processing*, 46(6):1688–1694, 1998.

[61] Andrea Soriano-Redondo, Ricardo A Correia, Vijay Barve, Thomas M Brooks, Stuart HM Butchart, Ivan Jarić, Ritwik Kulkarni, Richard J Ladle, Ana Sofia Vaz, and Enrico Di Minin. Harnessing online digital data in biodiversity monitoring. *Plos Biology*, 22(2):e3002497, 2024.

[62] Elias Sprengel, Martin Jaggi, Yannic Kilcher, and Thomas Hofmann. Audio based bird species identification using deep learning techniques. *LifeCLEF 2016*, pages 547–559, 2016.

[63] Robin Steenweg, Mark Hebblewhite, Roland Kays, Jorge Ahumada, Jason T Fisher, Cole Burton, Susan E Townsend, Chris Carbone, J Marcus Rowcliffe, Jesse Whittington, Jedediah Brodie, J Andrew Royle, Adam Switalski, Anthony P Clevenger, Nicole Heim, and Lindsey N Rich. Scaling-up camera traps: monitoring the planet's biodiversity with networks of remote sensors. *Frontiers in Ecology and the Environment*, 15(1):26–34, 2017.

[64] Larissa Sayuri Moreira Sugai, Thiago Sanna Freire Silva, José Wagner Ribeiro Jr, and Diego Llusia. Terrestrial passive acoustic monitoring: review and perspectives. *BioScience*, 69(1):15–25, 2019.

[65] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[66] Mathias W Tobler and George VN Powell. Estimating jaguar densities with camera traps: problems with current designs and recommendations for future studies. *Biological conservation*, 159:109–118, 2013.

[67] Luís Felipe Toledo, C Guilherme Becker, Célio FB Haddad, and Kelly R Zamudio. Rarity as an indicator of endangerment in neotropical frogs. *Biological Conservation*, 179:54–62, 2014.

[68] Michael Towsey, Birgit Planitz, Alfredo Nantes, Jason Wimmer, and Paul Roe. A toolbox for animal call recognition. *Bioacoustics*, 21(2):107–125, 2012.

[69] Hanna Tuomisto. What satellite imagery and large-scale field studies can tell about biodiversity patterns in amazonian forests. *Annals of the Missouri botanical garden*, pages 48–62, 1998.

[70] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, and Jonathan Bright. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[71] Kai Wang, Steven E Franklin, Xulin Guo, and Marc Cattet. Remote sensing of ecology, biodiversity and conservation: a review from the perspective of remote sensing specialists. *Sensors*, 10(11):9647–9667, 2010.

[72] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[73] Harry C Wilting, Aafke M Schipper, Michel Bakkenes, Johan R Meijer, and Mark AJ Huijbregts. Quantifying biodiversity losses due to human consumption: a global-scale footprint analysis. *Environmental science & technology*, 51(6):3298–3306, 2017.

[74] Christopher P Woods and R Mark Brigham. Common poorwill activity and calling behavior in relation to moonlight and predation. *The Wilson Journal of Ornithology*, 120(3):505–512, 2008.

[75] Hongyu Zhu, Mohamed Akrout, Bojian Zheng, Andrew Pelegris, Anand Jayarajan, Amar Phanishayee, Bianca Schroeder, and Gennady Pekhimenko. Benchmarking and analyzing deep neural network training. In *2018 IEEE International Symposium on Workload Characterization (IISWC)*, pages 88–100. IEEE, 2018.