Enabling Accessible and Adaptable AI for Bioacoustic Monitoring from Data Annotation to Edge Deployment

Santiago Martinez Balvanera

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Biosciences

University College London

28th March 2025

I, Santiago Martínez Balvanera, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Chapter 1: I was responsible for the conceptualisation, and writing the original draft. Kate Jones (KJ) and Oisin Mac Aodha (OMA) were responsible for supervision and reviewing and editing the manuscript.

Chapter 2: I was responsible for conceptualisation, investigation, software, visualisation, and writing the original draft. KJ and OMA were responsible for supervision, and reviewing and editing the manuscript. Holly Pringle (HP) and Matt Weldy (MW) were responsible for project administration, and validation. Ella Browning (EB), and MW also contributed to reviewing and editing the manuscript.

Chapter 3: I was responsible for conceptualisation, data curation, formal analysis, investigation, methodology, software, visualisation, and writing the original draft. Everardo Gustavo Robredo Esquivelzeta (EGRE) assisted with conceptualisation, data curation, and project administration. Veronica Zamora-Gutiérrez (VZG), Maria Cristina Mac Swiney Gonzalez (MCMSG) were responsible for data curation. KJ and OMA were responsible for supervision and reviewing and editing the manuscript.

Chapter 4: I was responsible for conceptualisation, data curation, formal analysis, investigation, methodology, software, visualisation, and writing the original draft. OMA was also responsible for conceptualisation, formal analysis, investigation, methodology, software, visualisation, and writing the original draft. OMA and KJ were responsible for supervision and reviewing and editing the manuscript. Elise Damstra (ED), Martyn Cooke (MC), Philip Eichinski (PE), Michel Barataud (MB), Katherine Boughey (KB), Roger Coles (RC), Giada Giacomini (GG), MCMSG, Martin K. Obrist (MO), Stuart Parsons (SP) and Thomas Sattler (TS) contributed with data curation. In particular, OMA conceived the project and conducted initial experiments training the model on four datasets (excluding Yucatan), leading the first draft of the manuscript. My contributions included expanding the code base for improved usability and training, annotating and contributing a new dataset, extending

and retraining all experiments with the addition of novel transfer learning experiments, performing performance analysis, and substantially revising the manuscript, including rewriting the introduction and abstract.

Chapter 5: I was responsible for conceptualisation, investigation, methodology, software, visualisation, and writing the original draft. Aude Vuilliomenet (AV) was also responsible for conceptualisation, investigation, methodology, software, visualisation and writing the original draft. KJ, OMA, and Duncan Smith (DS) were responsible for supervision, and reviewing and editing the manuscript. In particular, AV and I jointly conceived this project. I led the software design, with both of us contributing to its implementation. AV led the test deployments and drafted most of the initial manuscript, but I conducted a comprehensive rewrite.

Chapter 6: I was responsible for conceptualisation, and writing the original draft. I received comments on this chapter from KJ and OMA.

While I am the sole author of the text, except where noted, I acknowledge the use of ChatGPT 3.5 and Google Gemini Advanced for proofreading and minor edits to enhance clarity. All content and ideas remain my own.

Abstract

To address the critical challenge of biodiversity loss, it is essential to upscale monitoring efforts to help inform conservation actions. The growing application of AI for automated species detection and classification in audio streams offers a promising solution. However, the current application of AI in bioacoustics is limited in scope, often constrained by the lack of gold standard data, technical resource disparities, and a lack of accessible tools. In this thesis, I investigate techniques to facilitate accessible AI development and deployment in bioacoustics. Chapter 2 examines data annotation, the first stage of bioacoustic AI development, and whether current tools support collaborative and iterative improvement of AI models and datasets. I find that current bioacoustic annotation tools are insufficient for modern AI development and, in response, I develop `whombat`, an open-source tool designed for iterative data and model improvement. Chapter 3 investigates the type and quantity of annotations needed for effective bioacoustic classification. With an extensively annotated bat call dataset, I show that annotating spatio-temporal locations of calls substantially improves classification performance, especially in low-data scenarios. Chapter 4 investigates if standard AI techniques from computer vision are efficient for bioacoustic analysis. I show that models tailored to the temporal nature of bioacoustic data outperform previous approaches and adapt to small-scale bat call datasets from diverse regions. Chapter 5 examines deploying AI models on edge devices for bioacoustic monitoring, finding that while minimising maintenance and reliance on extensive data infrastructure, tailoring solutions to specific monitoring goals can require advanced coding skills. To address this challenge, I develop `acoupi`, an

open-source framework that simplifies the creation and deployment of such tailored solutions, with its effectiveness demonstrated through a month-long field deployment of a novel bat detection model. This research helps overcome challenges limiting AI-powered bioacoustics, paving the way for its broader use in conservation.

Impact Statement

This thesis advances AI-powered bioacoustic monitoring by providing practical guidance and developing accessible tools to facilitate its application in research and conservation. Tools presented are packaged into user-friendly open-source tools, helping democratise bioacoustic AI development and deployment.

In Chapter 2, I develop `whombat`, an open-source audio annotation tool designed to address a critical bottleneck in bioacoustic AI development: creating high-quality datasets for model training and evaluation. `whombat`'s user-friendly interface facilitates collaborative annotation and supports iterative improvement of AI models and datasets. Its impact is demonstrated through adoption by diverse research groups and conservation organisations, including supporting ongoing research on avian monitoring, gibbon detection, and bat social calls. Notably, the Bat Conservation Trust (BCT) is using it to improve models for national-scale monitoring, and Mexico's National Commission for Biodiversity (CONABIO) employs it for creating a Mexican bat call library. This work is published in *Methods in Ecology and Evolution* and has been presented at the International Bioacoustics Congress 2023 and Cornell University's *BioacousTalks* seminar series.

Chapter 3 provides practical guidance on annotating bioacoustic data for AI model development, specifically for bat call classification. This research demonstrates that annotating the timing and frequency ranges of individual vocalisations significantly improves model performance, especially when data is limited. This insight is valuable as bioacoustic data is often scarce, making investment in detailed annotation, when coupled with tools like `whombat`, a cost-effective strategy for enhancing model

accuracy.

In Chapter 4, I develop BatDetect2, a novel, deep learning model for bat call detection and classification. Trained on `whombat`-generated annotations, BatDetect2 outperforms previous approaches on diverse datasets. Its user-friendly codebase simplifies running the model on new data and is being trialled by BCT for their National Bat Monitoring Surveys (https://www.bats.org.uk/our-work/national-bat-monitoring-programme) and incorporated into the ecoSound-web platform (https://ecosound-web.de/ecosound_web/). Furthermore, the codebase facilitates training new models on different datasets, attracting significant interest for developing a European-specific model. BatDetect2 shows that incorporating fundamental bioacoustic principles into model architecture significantly enhances performance, a concept applicable to other taxa and bioacoustic tasks.

In Chapter 5, I develop `acoupi`, an open-source Python framework designed to simplify deploying AI models on edge devices for bioacoustic monitoring. `acoupi` empowers users to create smart bioacoustic devices that perform on-device data processing and analysis, transmitting only essential information, solving the problem of needing to transfer, store, and process large volumes of raw audio data. By enabling on-device processing and providing simplified mechanisms for deploying custom AI models and monitoring workflows, `acoupi` significantly lowers the barrier to entry for teams lacking specialised infrastructure or advanced coding skills. Furthermore, it supports near-real-time monitoring, which is crucial for time-sensitive applications such as the early detection of invasive species. `acoupi` is particularly well-suited for permanent monitoring stations and promises to be a key enabler for the broader adoption of AI-powered passive acoustic monitoring (PAM) systems across research, conservation, and commercial sectors. Early interest in `acoupi`, including potential collaborations with the University of Edinburgh's Soprano program (https://information-services.ed.ac.uk/iot/soprano-project), demonstrates its potential to significantly impact the field of bioacoustic monitoring.

Paper Declaration Form

The chapters of this thesis correspond to the following papers, which are either published or in preparation for publication.

Chapter 2: Martínez Balvanera, S., Mac Aodha, O., Weldy, M. J., Pringle, H., Browning, E., & Jones, K. E. (2024). Whombat: An open-source audio annotation tool for machine learning assisted bioacoustics. *Methods in Ecology and Evolution.* DOI: https://doi.org/10.1111/2041-210X.14468

Chapter 3: Martínez Balvanera, S., Robredo Esquivelzeta, E. G., Zamora-Gutiérrez, V., Mac Swiney Gonzalez, M. C., Mac Aodha, O., & Jones, K. E. Detailed Annotations Boost Classification Performance in Automated Acoustic Identification. Pre-print in preparation for submission. To be submitted to *Methods in Ecology and Evolution.*

Chapter 4: Mac Aodha, O., Martínez Balvanera, S., Damstra, E., Cooke, M., Eichinski, P., Browning, E., Barataud, M., Boughey, K., Coles, R., Giacomini, G., M. Mac Swiney G., M.C., Obrist, M.K., Parsons, S., Sattler, T. & Jones, K. E. (2022). Towards a general approach for bat echolocation detection and classification. bioRxiv, 2022-12. Joint first author with Mac Aodha, O. Preprint published at https://doi.org/10.1101/2022.12.14.520490 has been substantially modified for this thesis. To be submitted to *Methods in Ecology and Evolution.*

Chapter 5: Vuillioumenet, A., Martínez Balvanera, S., Mac Aodha, O., Jones, K. E., & Duncan, W. acoupi: An Open-Source Python Framework for Deploying Bioacoustic Deep Learning Models on Edge Devices Pre-print in preparation for submission. Joint first author with Vuillioumenet, A. To be submitted to *Methods in Ecology and Evolution.*

Acknowledgements

When I sat down, face-mask on, in front of my computer screen for my first supervisory meeting in January 2021, I was on a speeding train bound to London, all bags on me and ready to settle in but facing substantial uncertainty. On the other side were my supervisors, Kate and Oisin, who, initially startled from the noise and strange background, quickly embraced the situation with an earnest laugh and the warmest grin. They quickly established the tone for what would be an incredibly supportive and understanding supervisory environment. To Kate, I would like to express my thanks for all the incredibly detailed and precise feedback and the broad vision on science and communication and its role in conservation, for being a role model of a great scientist who pushes the boundaries but puts sensibility and compassion at the center, and of course for trusting in me. I would like to thank Oisin for his super quick, always practical, and impossibly positive feedback. I am most grateful for his selfless availability and for sharing his unique viewpoint at the intersection of deep learning research and application. I cannot thank them enough for their patience, guidance, and unwavering support throughout this journey.

I would not have been here if not for Everardo, Veronica, and Cristina, and colleagues at CONABIO. Vero and Cristina hurled me into the world of bats, provided full support for my work and gave me their warm friendship, thank you. To Everardo, I would like to express my deep thanks for all the lengthy discussions about ecology, everything and nothing, which broadened my understanding of what ecology and, more generally, living systems are. You remain a source of inspiration. I would like to thank everyone at CONABIO for spreading their passion for biodiversity

conservation and sharing with me the appreciation for the infinite variety and value of Mexican biodiversity.

The PhD has been a joy to share with fellow humans at the People and Nature Lab and UCL. I've had the pleasure of joining a great team of wonderful, thoughful and kind people. Thanks to Omi, Aude, Erin, Peggy, Holly, Jason, Ben, Ella, Simon, Gee, John, Juan, and Alastair for all the great times at the camp, throwing axes, and eating the lab-grown vegetables in the garden. It has been inspiring to grow as a researcher with all of you.

Living in a foreign country can be hard at times, but I have been privileged to find a great network of support with whom to enjoy life. In the strangest of times and places, I found a group that not only cushioned my arrival to foreign lands but became a close family for the 7 months I lived in Oxford. Thank you, Rodrigo, Blas, and Carlota, for those slow Sundays of infinite quesadillas and "tutoriales Gary." Thanks to my Mexican family in London—Ale, Argel, Manuel, Clau, Ani, Ilya, Elena—for all the parties, reunions, and picnics where we could all feel a bit closer to home. A particular, deep, deep thanks to Leo and Bato, who have been caring for me beyond expectations throughout these four years. I am absolutely honored and grateful for all the years living together. Thanks also to Andrea and Manu for the time spent together at Hickling House—albeit shorter, it was filled with valuable late-night conversations and great times sharing the house, meals, and life.

My roots extend across the ocean and are nurtured by the life-long relationships of friends and family. To my friends from a past life as a mathematics student—Alan, Daniel, Jaime, Omar, Pablote, Pablito, Rigel, Roman, Tania and Viri—thank you, I learned much more from our shared journey than from the lectures themselves. To Luis, Nuri, and Fermin, I would like to thank you for your unquestioning friendship, for the support you provide me when we see each other and which requires no sign when we don't. To my sister, for inspiring me to be critical and passionate, but humane and relaxed, to enjoy life and take a wild laugh. My wider family—grandparents, uncles, aunts and cousins—have always been incredibly supportive

and loving, thank you. And of course, I would like to thank my parents. They have given me every tool at their disposal for me to thrive and pursue my goals, and tended to me with great love and care. It is the seeds of their vision for kindness with nature that brought me here.

Finally, to Lucia, my partner in life, thank you. Being at your side throughout the tumultuous journey that has been my PhD has been the most enriching and gratifying part of it all. I thank you deeply for your support at all times and in all things.

Contents

List of Figures

List of Tables

Glossary

Artificial Intelligence The broad field of computer science dedicated to creating systems that can perform tasks typically requiring human intelligence. These tasks encompass learning, reasoning, problem-solving, perception, and natural language understanding. In this thesis, AI primarily refers to the automation of data analysis to extract meaningful information from raw data.. 26

Deep Learning A subfield of Machine Learning that utilises artificial neural networks with multiple layers (hence "deep") to learn complex patterns from data. These networks are loosely inspired by the structure and function of biological neural networks in the brain. Deep learning models often involve a significantly larger number of parameters compared to traditional statistical learning methods, enabling them to learn intricate representations of data. . 38

Machine Learning A subfield of Artificial Intelligence that focuses on the development of algorithms that enable computer systems to learn from data without being explicitly programmed. These algorithms improve their performance on a specific task through experience, typically in the form of data. Machine learning is inherently data-driven.. 26

Abbreviations

This document is incomplete. The external file associated with the glossary 'acronym' (which should be called `main.acr`) hasn't been created.

Check the contents of the file `main.acn`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "main"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "main"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

Chapter 1

Introduction

1.1 Biodiversity monitoring with passive acoustics

Biodiversity is essential to the stability and functioning of Earth's ecosystems (Tilman et al., 2014). However, evidence indicates a widespread decline in biodiversity and the crucial benefits it provides (Díaz et al., 2019; IPBES, 2019). The Living Planet Index indicates a 73% average decline in the relative abundance of monitored wildlife populations across terrestrial, freshwater, and marine systems (WWF, 2024). Human-induced pressures like land use change, over-exploitation, invasive species, and climate change are the primary drivers of biodiversity loss, and their impact is expected to continue (Tilman et al., 2017; Newbold, 2018). This crisis has spurred global initiatives, such as the Kunming-Montreal Global Biodiversity Framework (KM GBF), aimed at halting and reversing biodiversity loss while ensuring the sustainable use and management of nature's contributions to people (CBD, 2022). Achieving these goals requires a detailed understanding of biodiversity across all scales, its contributions to people, and how human activities are impacting both (Xu et al., 2021; Nicholson et al., 2021; Williams et al., 2020).

Effective monitoring provides the essential data needed to assess the state of biodiversity, track changes in species populations and ecosystems, and evaluate the effectiveness of conservation interventions (Gonzalez et al., 2023; Stephenson et al., 2022). This need for effective monitoring is underscored by emerging national

regulatory frameworks, including the UK's Environment Act 2021, which reflects England's legally binding commitment to the KM GBF with ambitious goals to protect and recover biodiversity (zu Ermgassen et al., 2021). Such policies require robust monitoring to assess biodiversity levels before, during, and after development to ensure compliance (Bull et al., 2019). However, designing effective monitoring programs can be challenging due to the diverse needs and applications of biodiversity data (Sparrow et al., 2020). For instance, long-term, large-scale monitoring is crucial for understanding baseline ecosystem structure and function and for detecting deviations from healthy or desired states (Likens & Lindenmayer, 2018). Rapid monitoring is key to enabling swift intervention for mitigating the negative outcomes of increasing human-wildlife interactions (Nyhus, 2016), such as the intrusion of invasive species (Martinez et al., 2020) and poaching of protected species (Kamminga et al., 2018). Furthermore, monitoring efforts require transparency and reproducibility to be effectively tracked, audited, and aligned with best practices (Bull et al., 2019). To achieve a comprehensive understanding of biodiversity across all scales, monitoring data must be integrable, highlighting the need for standardised and shared practices that enable effective analysis and comparison (Schmeller et al., 2015). These diverse and demanding requirements underscore the importance of innovative monitoring approaches that can provide reliable, scalable, and timely data to support effective conservation action (Besson et al., 2022; Stephenson et al., 2022).

Passive Acoustic Monitoring (PAM) offers a promising approach to large-scale biodiversity monitoring (Gibb et al., 2018). By deploying networks of acoustic sensors to record soundscapes, PAM enables non-invasive monitoring of a diverse range of taxa, including elusive species that are difficult to detect visually, such as those inhabiting dense vegetation or exhibiting nocturnal behaviour. Acoustic signals provide information on a wide range of taxa, encompassing terrestrial fauna such as bats (Milchram et al., 2020; Reichert et al., 2021), birds (Sethi et al., 2024), insects (Riede & Balakrishnan, 2024; Ganchev et al., 2007), anurans (Melo et al., 2021; Lapp et al., 2021), elephants (Wrege et al., 2017), primates (Kalan et al.,

2015), as well as aquatic organisms, including marine mammals (Cauchy et al., 2020; Kowarski & Moors-Murphy, 2020) and freshwater insects (Desjonquères et al., 2024). Furthermore, PAM can capture anthropogenic sounds such as gunshots (Hill et al., 2018; Wijers et al., 2019) and chainsaws (Somwong et al., 2023) providing insights into human-wildlife interactions and a measure of human pressure on the environment (Fairbrass et al., 2019). These rich acoustic data can be used to generate or complement biodiversity assessments (Gasc et al., 2013; Zwerts et al., 2021; Hoefer et al., 2023), study soundscape patterns linked to ecosystem diversity and complexity (Alcocer et al., 2022), track ecological recovery and disturbance (Znidersic & Watson, 2022) in various ecosystems including reefs (Lamont et al., 2021) and forest soil (Robinson, Breed et al., 2023), and advance fundamental ecological research (Ross et al., 2023). PAM deployments operate autonomously over extensive areas and extended periods, facilitating continuous and widespread monitoring without constant supervision. Recent technological advancements have reduced the costs of acoustic sensors (Hill et al., 2019; Sethi et al., 2018; Lamont et al., 2022), facilitating large-scale deployments even under constrained budgets (Williams et al., 2018). Recorded audio can be stored for later analysis, allowing for data validation and re-analysis. As a result, PAM is increasingly employed for biodiversity assessments and fundamental research in ecology and conservation science (Sugai et al., 2018).

1.2 Bioacoustic detection using artificial intelligence

Efficiently identifying animal vocalisations within PAM recordings is essential for understanding soundscapes. While broader soundscape analyses can estimate faunal diversity or predict species presence indirectly (Sethi et al., 2022; Bradfer-Lawrence et al., 2019; Bradfer-Lawrence et al., 2024), they are susceptible to confounding factors, such as sounds from human activity or climatic events like rain and wind, leading to inconsistent assessments of acoustic diversity (Ross et al., 2021; Alcocer et al., 2022; Fairbrass et al., 2017). In contrast, the ability to detect sounds from target species provides a clearer and more interpretable understanding of the soundscape.

However, PAM deployments often result in vast audio datasets, sometimes reaching tens of millions of hours (Roe et al., 2021), with key vocalisations hidden amongst the background ambient noise. Manual identification is constrained by the limited availability of qualified experts and is impractical for large-scale PAM efforts (Fraser, 2018). Artificial Intelligence offers a powerful solution by automating this task, enabling scalable and efficient analysis of PAM datasets (Tuia et al., 2022; Besson et al., 2022; Farley et al., 2018; Christin et al., 2023; Pichler & Hartig, 2023). AI systems for bioacoustics are primarily implemented using Machine Learning, a data-driven approach where models, guided by expert-annotated training data, learn to detect and classify acoustic events directly from audio recordings (Pichler & Hartig, 2023). This approach can automatically identify and utilise discriminative features that may be difficult to articulate algorithmically (Borowiec et al., 2022) or that may have been overlooked due to human perceptual biases (Kershenbaum et al., 2014). AI-driven systems, when aligned with open science principles (Hampton et al., 2015), promote reproducible analysis and ensure consistent processing, reducing inter-observer variability (Farmer et al., 2012), and enabling re-assessment under novel conditions (Wood & Kahl, 2024). Furthermore, AI algorithms can be deployed across diverse computing environments, from centralised cloud infrastructure (Sethi et al., 2020), to local machines like laptops and workstations (Kahl et al., 2021) and, increasingly, resource-constrained edge devices (Sheng et al., 2019; Höchst et al., 2022), enabling diverse workflows and allowing for near-real-time detection. By automating the detection of bioacoustic signals, AI is rapidly becoming an indispensable tool for PAM-based biodiversity monitoring (Sharma et al., 2022).

Despite the increasing use of AI in bioacoustics, the field has yet to fully capitalise on its potential for comprehensive and accessible monitoring. Existing AI models cover only a small fraction of vocalising species and are confined to regions with abundant, readily available data (Gibb et al., 2018; Nieto-Mora et al., 2023). For example, BirdNET (Kahl et al., 2021), a leading AI model for bioacoustic identification, can nominally identify approximately 6,000 bird species ($\sim$56% of known species), but its performance has been rigorously assessed for only around 1,000

species (<10%), mostly from Europe and North America (Pérez-Granados, 2023). Bats, with approximately 1,100 echolocating species, represent another highly diverse group readily detectable through acoustic monitoring (Jones & Teeling, 2006; Jakobsen et al., 2013). However, despite the development of AI-powered bat call detecting algorithms (Mac Aodha et al., 2018; Kobayashi et al., 2021; Paumen et al., 2021; Zualkernan et al., 2020; Chen et al., 2020; Zhang et al., 2021; Khalighifar et al., 2022; Vogelbacher et al., 2023; Yoh et al., 2022; Fundel et al., 2023; Alipek et al., 2023), these tools currently cover around 120 species with most lacking public accessibility and code for use (only 4 out of 11 studies provide this, also see Baker & Vincent, 2019). Ironically, such bioacoustic AI models are most lacking in the world's most biodiverse areas, including tropical regions facing increasing human pressures (Newbold et al., 2020) and severe data deficiency (Frick et al., 2019; Collen et al., 2008). The gap limits the establishment of comprehensive and unbiased monitoring programs, potentially leading to severe consequences for conservation efforts in these critical areas. However, developing AI for bioacoustics presents significant challenges, from data collection and curation to model training, evaluation, and real-world deployment. Fully harnessing the potential of AI for biodiversity monitoring and conservation requires prioritising research and development focused on efficient model creation and adaptation for broader applicability.

1.3 Bioacoustic AI through expert annotation

High-quality data is the foundation for developing AI models for acoustic species detection in bioacoustics (Chasmai et al., 2024). These data comprise recordings, each labelled to indicate the species vocalising, providing the AI models with examples of the species' sounds for training. By learning from diverse examples of both target and non-target sounds, the model discerns discriminative patterns in the target vocalisations, enabling it to identify the species in novel recordings (Borowiec et al., 2022). To evaluate model performance, these recordings are also used to compare model predictions with known vocalisations (Mesaros et al., 2021; van Merriënboer et al., 2024). Ideally, these recordings should encompass diverse acoustic condi-

tions, reflecting the variety of environments where the model will be deployed (van Merriënboer et al., 2024). However, acquiring such comprehensive data presents a significant challenge (Pichler & Hartig, 2023). Bioacoustic fieldwork typically involves complex logistics, extended observation periods, and can necessitate the capture of individuals to confirm species identification (Zamora-Gutierrez et al., 2020). Self-supervised learning, where AI models learn patterns within the data itself, offers a promising approach to reduce reliance on labelled data (Liu et al., 2022). However, these methods underperform when distinguishing between subtle classes (Cole et al., 2022), a common challenge in bioacoustics (Chasmai et al., 2024). This situation underscores the critical importance of fully leveraging existing labelled bioacoustic datasets.

The utility of existing bioacoustic data can be significantly enhanced through careful manual expert-review and annotation of audio recordings. Bioacoustic datasets often comprise lengthy focal recordings targeting individual species, but also capturing extraneous sounds like background noise and vocalisations from other animals (Hamer et al., 2023). Knowing that a species vocalises within a recording but not the precise timing of their vocalisations results in "weak labels," making AI model training more difficult. To train acoustic identification models, shorter clips are commonly extracted from longer recordings to provide examples of the recorded species (Stowell, 2022). However, weak labelling makes it difficult to ensure these clips contain only the target sound, potentially introducing irrelevant noise that confuses the AI model (Shah et al., 2018). Therefore, a crucial step is to manually review and annotate the audio material, precisely marking the timing of all relevant sounds to generate "strong labels." Although AI models can be trained with weak labels (Kong et al., 2019; Kumar & Raj, 2016), Hershey et al., 2021 found that even simple "strong labels" specifying only event onset and offset times improved model performance, suggesting that more detailed annotation could yield further gains. Additionally, annotations enable detailed analysis of model errors, such as missed detections of non-focal species or misclassifications, aiding in diagnosing performance issues (van Merriënboer et al., 2024). Despite these

benefits, comprehensive annotation of acoustic datasets remains rare (Chasmai et al., 2024), representing a significant untapped opportunity to enhance AI performance in bioacoustics.

1.4 Bioacoustic annotation for AI development

Annotating bioacoustic datasets is a complex process demanding meticulous effort and considerable time (Cartwright et al., 2019; Fraser, 2018). Accurately identifying and labelling target sounds within a recording requires careful aural and visual inspection (Fraser, 2018), typically utilising spectrograms or other visual representations tailored to the specific acoustic characteristics of the target species or sounds (Odom et al., 2021). This includes discerning faint or masked vocalisations, which pose significant challenges for AI models (Stowell, 2022) and necessitate thorough examination to ensure comprehensive annotation. While species identification is common, capturing additional information like vocalisation structure, function (e.g., alarm calls, territorial defence), and biological context (e.g., life stage, sex, individual) can significantly benefit both model training and downstream analysis, as intra-species vocalisations can vary widely (Teixeira et al., 2019; Odom et al., 2021). Ideally, a standardised taxonomy or ontology of labels should be used to capture all possible information and facilitate data sharing and collaboration, though this presents challenges for consistent label management (Roch et al., 2016). Annotators may want to label non-target sounds as well, as these can provide valuable context for understanding model performance or address other research questions. Because annotation results can vary between individuals, multiple experts should ideally review the same recordings to ensure the quality and validity of the annotations (Nguyen Hong Duc et al., 2021). Consequently, annotation projects, particularly those requiring multiple annotators, varied target sounds, and rich metadata, face significant logistical and methodological challenges that must be managed carefully to ensure data quality and consistency.

Specialised software tools play a key role in streamlining the annotation process. Popular choices include versatile audio analysis tools like Raven and Audacity (Con-

servation Bioacoustics, 2023; Audacity, 2017), and dedicated software with a focus on specific regional fauna, such as Marsland et al. (2019) with a particular focus on New Zealand bird species. However, these tools fall short in supporting the iterative nature of AI model development. Building successful models requires continuous refinement of both the data and the model itself (Zha et al., 2023; Jarrahi et al., 2022). For example, addressing annotation quality issues can yield significant improvements in model performance (Budach et al., 2022), while analysing performance gaps can guide targeted data annotation or collection efforts (Roscher et al., 2024). Furthermore, real-world deployments typically encounter evolving data distributions, such as when models are applied to new locations or when seasonal shifts alter the acoustic environment (Bidarouni & Abeßer, 2024). These dynamic conditions necessitate ongoing data annotation and curation to maintain and improve model accuracy (Rabanser et al., 2019). Unfortunately, the current generation of software tools for audio annotation fails to facilitate this iterative process and may not adequately support the specific needs of AI development for bioacoustics.

1.5 Detailed annotations and their application in bioacoustic AI

Even with the best tools, annotating data is not a straightforward or standardised process. Due to the time and expense involved, efforts are often made to reduce or optimise the amount of annotation required (McEwen et al., 2024; Tejero et al., 2023). One common approach is to segment the recordings in the dataset into shorter clips and manually identify those containing target sounds (Hershey et al., 2021; Khalighifar et al., 2022). This method can be quick, as it only requires identifying the presence of the target sound within a clip. However, the resulting annotations can be coarse, presenting similar challenges to weak labels, such as the inclusion of extraneous sounds and noise, particularly when clips are much longer than the target sounds. Other approaches, such as precisely marking the start and end times of each target sound or delineating its frequency range, are more time-consuming but offer greater resolution and accuracy (Morfi et al., 2019; Cañas et al., 2023;

Lostanlen et al., 2018). While large-scale datasets may necessitate faster annotation approaches, bioacoustics often involves limited recordings per species (Nolasco, Singh et al., 2023; Nolasco et al., 2022). In these cases, optimising annotation procedures becomes essential, as improving existing data may be more feasible than collecting new examples. However, it remains unclear which annotation approach is most effective and how to best balance time, effort, annotation quality, and the resulting impact on model performance.

Annotated bioacoustic data offers a rich source of information beyond simple identification of target vocalisations. The precise timing, frequency, and structure information embedded within annotations can be leveraged to enhance AI training, moving beyond simply identifying which clips contain sounds. This granular information is critical because models trained on limited bioacoustic datasets are susceptible to overfitting (Wei et al., 2020), whereby they leverage spurious correlations (e.g., background noise) rather than learning genuine vocalisation features (Ying, 2019), hindering generalisation to novel soundscapes. Strategies such as multi-task learning (Zhang & Yang, 2022; Martin et al., 2022; Morfi & Stowell, 2018), where a model is trained to concurrently perform multiple distinct tasks, have been explored to mitigate overfitting and promote robust learning. This encourages the model to learn features relevant to all tasks, thereby improving generalisation and leading to improved performance (Standley et al., 2020). Applying this strategy to bioacoustics could involve training a model to both classify and locate sound events within spectrograms, thereby leveraging the detailed information available in annotations. Another promising avenue involves shifting from simply identifying the presence of vocalisations within an audio clip to directly predicting the precise location of each sound event using annotations as training targets (Venkatesh et al., 2022), analogous to object detection methods in computer vision (Beery et al., 2019; Zou et al., 2023). This strategy not only has the potential to improve model performance but also facilitates more granular analyses of animal communication, enabling investigations into call sequences or variations in finer call structure. Despite the potential of these approaches, their application and efficacy in bioacoustics remain

largely underexplored (Stowell, 2022).

1.6 Overcoming data scarcity in bioacoustic AI training

Another key challenge in bioacoustics is identifying AI models and architectures that effectively leverage the distinctive characteristics of bioacoustic signals to maximise performance. This is particularly important because they are commonly developed with limited data (Nolasco, Singh et al., 2023), even when meticulous annotation is employed. Bioacoustic signals can be remarkably subtle, as species vocalisations may only be distinguishable by small variations in frequency or temporal patterns (Odom et al., 2021). For example, identifying bat species often requires analysing entire call sequences (Russ, 2021), whereas subtle variations in syllable repetition can distinguish birdsong between species (Dalziell et al., 2014). Currently, most AI development for bioacoustics adapts techniques from computer vision (Stowell, 2022), relying on the assumption that visual patterns in spectrograms, or other image-based representations of audio, are sufficient for distinguishing between species. While successful with large and diverse datasets (Kahl et al., 2021; Ghani et al., 2023; Kong et al., 2020), these methods typically require significant modification for effective application with limited training examples (Nolasco et al., 2022; Nolasco, Singh et al., 2023; Nolasco, Ghani et al., 2023). This raises the question of whether this approach is truly efficient, or if alternative methods, specifically designed for audio, could encode bioacoustic patterns more effectively and require less training data. While some studies have explored recurrent neural networks (Madhusudhana et al., 2021; Gupta et al., 2021) and transformers (Fundel et al., 2023) to address the temporal structure of audio, these approaches still rely on visual models. Directly processing raw audio with models like SincNet (Bravo Sanchez et al., 2021), a 1-dimensional convolutional neural network, has shown comparable performance with fewer parameters, but has not yet surpassed visual approaches. Given the promising results of initial explorations with raw audio processing, further investigation into audio-specific models is crucial for maximising the efficiency of limited data.

Leveraging existing data, even if not directly relevant to the specific task, can help alleviate the challenges posed by scarce training data. Transfer learning is a common technique that allows models to leverage knowledge from a source task to improve performance on a target task (Kong et al., 2020; Zhuang et al., 2021; Tsalera et al., 2021). For instance, pre-training large models on existing data from other taxonomic groups and then fine-tuning them for a specific task has proven effective in bioacoustics (Ghani et al., 2023; Dufourq et al., 2022; Williams et al., 2024). Other approaches utilise multi-modal data, such as text or images, to create models capable of learning from multiple sources of information, with the aim of improving audio understanding (Robinson, Robinson et al., 2023; Miao et al., 2023). These techniques hold promise for accelerating the development of AI bioacoustic models, especially in areas or tasks where data is scarce. However, most transfer learning efforts in bioacoustics focus on transferring knowledge between distinct taxonomic groups. While sufficient data may exist for common sound types within a specific taxonomic group like bats (Roemer et al., 2021), significant species and regional gaps often persist (Frick et al., 2019). For example, despite existing datasets for certain regions (Görföl et al., 2022; Khalighifar et al., 2022; Vellinga & Planque, 2015; Zamora-Gutierrez et al., 2020), transferring knowledge to understudied areas with potentially different acoustic characteristics presents a challenge. Even if existing data encompasses all call types (Roemer et al., 2021), regional variations in vocalisations and background noise can hinder the effectiveness of models trained on data from different locations (Russo et al., 2018). This highlights the need for research into effective transfer learning strategies within taxonomic groups, particularly for species with limited data and geographically diverse vocalisations.

1.7 On-device bioacoustic analysis

Moving from AI model development to real-world acoustic monitoring introduces a distinct set of challenges beyond those encountered in the initial development phase. The large quantities of data generated by large-scale monitoring efforts, especially in remote areas, pose significant challenges for data management, processing, and

transfer (Browning et al., 2017). Regular manual intervention for data retrieval is commonplace in current monitoring efforts (Roe et al., 2021), and data storage can quickly become problematic for large, long-term deployments (Kowarski & Moors-Murphy, 2020). Furthermore, analysing this data with AI models demands significant computational resources, and processing large datasets often requires specialised infrastructure (Sethi et al., 2018). Consequently, while bioacoustic methods hold significant appeal, their practical application is often limited by these challenges.

Edge computing offers a compelling solution to data management and post-processing challenges. In edge computing, data processing occurs directly on devices in the field, enabling them to run AI models locally and transmit only the results, thereby significantly reducing data transfer and storage needs (Shi et al., 2016). Recent technological advancements have enabled the development of affordable devices capable of recording, processing, and transmitting data via cellular or low-power Long-Range Wide-Area Network (LoRaWAN), facilitating near real-time analysis (Sethi et al., 2018; Gallacher et al., 2021; Baucas & Spachos, 2020). While requiring additional computational and power resources, these devices are particularly well-suited for monitoring stations with access to reliable power sources, such as solar panels, and consistent network connectivity. Ongoing research into computationally efficient models (Höchst et al., 2022; Surianarayanan et al., 2023) and power-saving strategies (Millar et al., 2024) for edge devices further enhances the appeal of this approach.

Networks of smart devices have been successfully deployed for various monitoring applications, such as monitoring bird populations across Norway (Bick et al., 2024) and identifying wolves to manage increasing human-wolf conflict in the Alps (Stähli et al., 2022). The popular BirdNET model, adapted for the Raspberry Pi—a popular platform for edge computing (Jolles, 2021)—feed citizen-based networks that provide real-time bird detections worldwide through platforms like BirdWeather (Clark et al., 2023). However, significant challenges remain in bringing AI to the edge. Firstly, developing devices robust enough for reliable field operation presents a complex engineering challenge. Secondly, existing edge systems, often

designed for specific applications, lack the flexibility to readily adapt to new AI models or data collection strategies. Consequently, the limited adaptability of existing edge systems restricts the immediate application of bioacoustic AI models across the varied landscape of acoustic monitoring.

1.8 Thesis overview

In this thesis, I explore and develop novel methodologies to accelerate and adapt the development and application of bioacoustic AI models for biodiversity monitoring. While these methods are taxon-independent, I focus on bats as a case study due to the unique challenges they present for acoustic identification. The methodologies here explored encompass key stages of the AI-assisted bioacoustic monitoring pipeline, including data annotation, model development, and model deployment.

In Chapter 2, I investigate current annotation practices in bioacoustics and advocate for a data-centric approach to AI model development. A review of existing annotation tools reveals that none fully support the iterative workflow required for effective AI development. To address this gap, I develop `whombat`, a novel, open-source software tool specifically designed to streamline bioacoustic annotation workflows. This user-friendly tool aims to empower researchers to curate, grow, and maintain the data necessary for robust AI model development.

In Chapter 3, I investigate diverse approaches to annotating the presence of relevant sound events within long audio recordings. Using a richly annotated dataset of bat echolocation calls, I simulate the training of AI detectors and classifiers using various annotation approaches and evaluate their performance on a challenging, held-out test set. When detailed annotations are available, I augment the training regime to include a time-frequency localisation task in a multi-task learning framework, thereby exploiting the richer information provided. This training is conducted across scenarios with varying amounts of training data to analyse how performance is impacted by annotation approach and training dataset size.

In Chapter 4, I introduce a novel architecture for bat detection and classification.

This architecture incorporates modifications that enable the model to reason explicitly across longer timescales and integrate frequency information from the spectrogram, diverging from traditional visual models. Leveraging detailed annotations, the model is trained to jointly locate each call within the spectrogram using a bounding box and predict its species. To assess its adaptability, I train the model on datasets from diverse geographic regions and conduct experiments to evaluate its transferability between these regions. This model and the associated training approach are made available as an open-source tool to facilitate broader adoption.

In Chapter 5, I address the challenges of edge processing and adapting existing systems to utilise novel AI bioacoustic models. Recognising that current systems lack adaptability, I develop and present an open-source framework designed to facilitate the development and deployment of edge devices for bioacoustics. This framework aims to empower researchers and hobbyists to customise and deploy acoustic monitoring stations tailored to their specific monitoring requirements.

Finally, Chapter 6 discusses limitations of the proposed methods and significant challenges in the field, whilst outlining future research avenues for achieving scalable passive acoustic monitoring in practice.

Chapter 2

Whombat: An Open-Source Annotation Tool for Machine Learning Development in Bioacoustics

2.1 Abstract

Automated analysis of bioacoustic recordings using Deep Learning (DL) methods has the potential to greatly scale biodiversity monitoring efforts. The use of DL for high-stakes applications, such as conservation and scientific research, demands a data-centric approach with a focus on selecting and utilising carefully annotated and curated evaluation and training data that is relevant and representative. Creating annotated bioacoustic datasets presents a number of challenges, such as managing large collections of recordings with associated metadata, developing flexible annotation tools that can accommodate the diverse range of vocalisation profiles of different organisms, and addressing the scarcity of expert annotators. Here I develop `whombat`, a user-friendly, browser-based interface for managing audio recordings and annotation projects, with several visualisation, exploration, and annotation tools. It enables users to quickly annotate, review, and share annotations, as well as visualise and evaluate a set of DL predictions on a dataset. The tool facilitates an iterative workflow where user annotations and DL predictions feed back to enhance model

performance and annotation quality. I demonstrate the flexibility of `whombat` by showcasing two distinct use cases: (1) a project aimed at enhancing automated UK bat call identification at the Bat Conservation Trust (BCT), and (2) a collaborative effort among the USDA Forest Service and Oregon State University researchers exploring bioacoustic applications and extending automated avian classification models in the Pacific Northwest, USA. `whombat` is a flexible tool that can effectively address the challenges of annotation for bioacoustic research. It can be used for individual and collaborative work, hosted on a shared server or accessed remotely, or run on a personal computer without the need for coding skills.

2.2 Introduction

Recent advancements in Deep Learning are revolutionising our ability to analyse large datasets generated by passive acoustic recorders for ecologically relevant signals (Kitzes et al., 2021; Tuia et al., 2022). Open-source Deep Learning models, such as BirdNET (Kahl et al., 2021) and NABat ML (Khalighifar et al., 2022), can be used to monitor birds and bats at scale across large regions. While considerable attention has been directed towards developing sophisticated DL systems, it is crucial to acknowledge the pivotal role of data and the various tasks encompassed within data work in establishing reliable DL implementations (Sambasivan et al., 2021). These tasks include Discovery, Capture, Curation, Design, and Creation of data which collectively contribute to the quality and effectiveness of DL models (Muller et al., 2019). In line with this, the data-centric approach has gained increasing relevance (Jarrahi et al., 2022), emphasising the collection, curation, and management of high-quality training and evaluation data to comprehensively assess model performance and ensure reliability, particularly in high-stakes applications such as conservation. Data work is inherently complex, and audio annotation, encompassing the identification of the location of relevant sound events in audio recordings and the assignment of appropriate labels, represents a time-consuming and labour-intensive process (Cartwright et al., 2019). Often, the creation of DL-ready datasets relies on software tools and technical infrastructure to ease management and enhance

efficiency (Reichert et al., 2021; Roe et al., 2021). However, while the broader DL community has recognised the importance of providing accessible, efficient and open-source tools for dataset curation and annotation (Sager et al., 2021; Neves & Seva, 2020), the bioacoustics community has lagged behind (Stowell, 2022; Tuia et al., 2022).

The annotation process is an integral part of an iterative workflow aimed at continually improving and monitoring the performance of DL models and data quality (Hohman et al., 2020). The evaluation of DL models can help identify errors and areas for potential improvement, such as annotation or data gaps, thereby increasing confidence in the performance of the model (Nahar et al., 2022). Continual annotation of novel data is crucial to monitor the performance of DL models, particularly when exposed to unknown environments, as these can pose a risk to model accuracy and reliability (Saria & Subbaswamy, 2019). However, existing annotation tools often lack appropriate design for effective annotation and DL development, hindering the seamless execution of this valuable feedback loop (Table 2.1).

Table 2.1: Comparison of seven popular software used for acoustic annotation. Dashes indicate that the corresponding feature is not supported by the software (to the best of the authors' knowledge), while a checkmark indicates its availability (see Appendix A.1 for further details).

	`whombat`	Arbimon[1]	AvianZ[2]	Kaleidoscope[3]	Label Studio[4]	Raven[5]	Sonic Visualiser[6]
Open-source	✓	–	✓	–	✓	–	✓
Self-Host	✓	–	✓	✓	✓	✓	✓
Collaborative	✓	✓	–	–	✓	–	–
Large Datasets	✓	✓	✓	–	✓	–	–
Rich Metadata	✓	✓	–	✓	–	–	–
Search Capabilities	✓	✓	–	–	–	–	–
Annotation Exploration	✓	✓	–	–	–	–	–
Flexible Spectrogram	✓	–	✓	✓	–	✓	✓
Flexible Annotations	✓	✓	✓	–	–	–	–
Quality Assurance	✓	✓	–	–	–	–	–
Training Tools	✓	–	–	–	–	–	–
Prediction Evaluation	✓	✓	✓	–	–	–	–
Export Annotations	✓	–	–	–	✓	✓	✓
Integrated Detectors	–	✓	✓	✓	–	–	–

Creating DL-ready datasets for bioacoustic research is a collaborative effort (Zhang et al., 2020) that requires a combination of modelling, analysis, annotation work, and quality assurance (Jarrahi et al., 2022; Muller et al., 2019). Annotation

can be accelerated if tackled by teams working simultaneously and distributing the workload among members with specialised and expert knowledge (Muller et al., 2021; Cartwright et al., 2019). However, managing large collections of audio recordings in bioacoustic research can be overwhelming (Kvsn et al., 2020), as they often contain hundreds or thousands of recordings (Zhang et al., 2013), each with its own set of metadata such as location, date, and time of recording, as well as other relevant contextual information. Storing the associated metadata is desired as it can influence modelling decisions and provide contextual cues for acoustic identification (Kshirsagar et al., 2021; Paullada et al., 2021). Being able to locate specific recordings or annotations within these collections is crucial for effective analysis and research but can be time-consuming and difficult without proper tools (Kandel et al., 2012). Providing a platform for collaborative annotation requires finding a balance between accessibility, simplicity, and the ability to manage complex and diverse workflows (Simpson et al., 2014).

Bioacoustic annotation is a challenging task due to the wide variety of organisms and vocalisation profiles that are studied in bioacoustic research (Stowell, 2022; Odom et al., 2021). Some animals produce long duration and broad-band sounds, while others produce vocalisations that can be clearly localised both in time and frequency. Substantial expertise in the acoustic identification of the target animal is often required and acquiring this knowledge can be a challenging process, often requiring extensive field experience. The pool of bioacoustic experts per taxon is, therefore, typically small and their expert annotation time is valuable (Nahar et al., 2022). While existing annotated data can serve as valuable reference material for training, the process of upskilling annotators often requires structured guidance and a systematic presentation of diverse target sounds. Existing annotation tools, though possessing many components suitable for training, lack features specifically tailored for this purpose. Additionally, in order to effectively accommodate the varying characteristics of different types of biological sounds, annotation tools must be flexible in terms of their visualisation and annotation capabilities (Stowell, 2022). Furthermore, generic audio annotation tools are primarily focused on the analysis of

human speech or music and lack the necessary visual representation of audio and consideration of recording context. Conversely, specialised bioacoustic software has often focused on specific taxonomic groups (Szewczak, 2010; Marsland et al., 2019), making it difficult to use these tools for the analysis of other groups. Despite the availability of a variety of annotation tools, none have been able to fully address the complexity of challenges that are inherent to bioacoustic research (Table 2.1; see Appendix A.1 for a thorough evaluation of existing audio annotation tools).

Here I develop `whombat`, a flexible tool specifically designed to accelerate bioacoustic DL research by facilitating the curation of annotated acoustic datasets. `whombat` offers a user-friendly browser-based interface that enables efficient management of acoustic datasets and annotation projects. It provides various visualisation, exploration, and annotation tools that allow users to annotate, review, and share annotations with ease. Moreover, these exploration tools can be employed to visualise, evaluate, and explore DL predictions on annotated datasets. `whombat` supports an iterative workflow (Figure 2.1), where user annotations and DL predictions continuously enhance both model performance and annotation quality. Additionally, `whombat` is designed to support both individual and collaborative work, enabling hosting on shared servers, cloud platforms, or private premises with remote accessibility. Notably, it can also run on personal computers without internet access. The application code is open-source and available at `https://github.com/mbsantiago/whombat`. To ensure accessibility for all users, I have bundled the tool into executable files for Windows, macOS, and Ubuntu, eliminating the need for dependency installation or coding skills. By making `whombat` open-source and easily accessible, I aim to empower researchers in bioacoustic DL research and foster advancements in the field.

2.3 Software Features

In this section I provide a brief description of the features and interface of `whombat`, following the order of the intended annotation workflow (Fig. 2.1). This includes the initial setup and loading of data, visualisation and navigation tools, annotation cap-

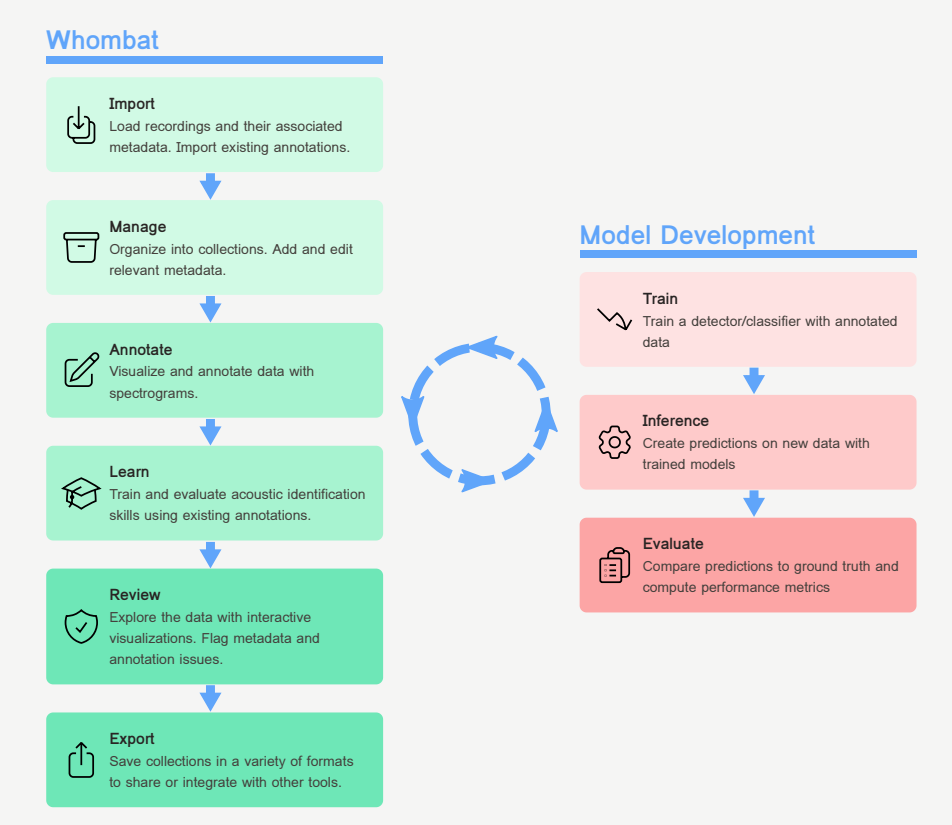


Figure 2.1: Iterative workflow of the `whombat` annotation tool. The iterative workflow of the `whombat` annotation tool. The application enables a feedback loop between user annotations and machine learning predictions, enhancing both model performance and annotation quality. The capabilities of the tool are represented by the green boxes on the left, while the red boxes on the right illustrate the steps in the model development workflow. The arrows indicate the typical direction of the workflow, but the tool provides flexibility for users to navigate between steps. Dashed arrows indicate potential crossover between model and data development. `whombat` allows users to export annotations for training machine learning models and then import predictions for comparison with existing annotations. This two-way flow of information empowers users to explore and integrate both components for their analysis.

abilities, quality control features, and DL model evaluation. Through this overview, I demonstrate how `whombat` can enhance the efficiency and accuracy of bioacoustic annotation.

2.3.1 Dataset management

The workflow begins by creating a acoustic dataset (Fig. 2.1). A dataset can be created by selecting all recordings within a folder or by importing a pre-existing dataset. The tool supports various audio file formats, including popular lossless formats in Bioacoustics such as WAV and FLAC, as well as the lossy MP3 format and others. Multiple datasets can be managed simultaneously.

Basic media information is scanned and stored for each recording, including its duration, number of channels, and sample rate (Fig. 2.2). `whombat` also allows the retrieval of metadata from commonly used autonomous recording units, e.g. Wildife Acoustics and AudioMoth (Hill et al., 2019). Users can edit the location and date-time of recordings on a per-recording basis or import this information from CSV files. Additionally, recordings can be tagged with multiple key-value pairs, providing contextual information relevant to the annotation process. For example, a recording can be tagged with key-value pairs like `species:Myotis lucifugus`, `sex:Male`, `age:Adult`, and `habitat:Forest`, to describe the recording target and context. In essence, a key-value pair is a simple way to store data where one piece of information acts as a label (key) and another piece holds the corresponding value. Here, 'species' is the key, and *Myotis lucifugus* is the value associated with that key. This approach allows for flexible, organised, and extensible metadata management.

To explore datasets, users can listen to recordings and visualise their spectrograms. `whombat` uses spectrograms as the main visualisation tool, as they facilitate the quick identification of sound events (Cartwright et al., 2019). Spectrogram parameters and other visual settings are configurable to best suit target sounds. `whombat` dynamically generates spectrogram sections on the fly, optimising computational efficiency and preventing excessive memory usage for long recordings. This allows for easy navigation using scroll bars, eliminating the need to compute and store large spectrograms

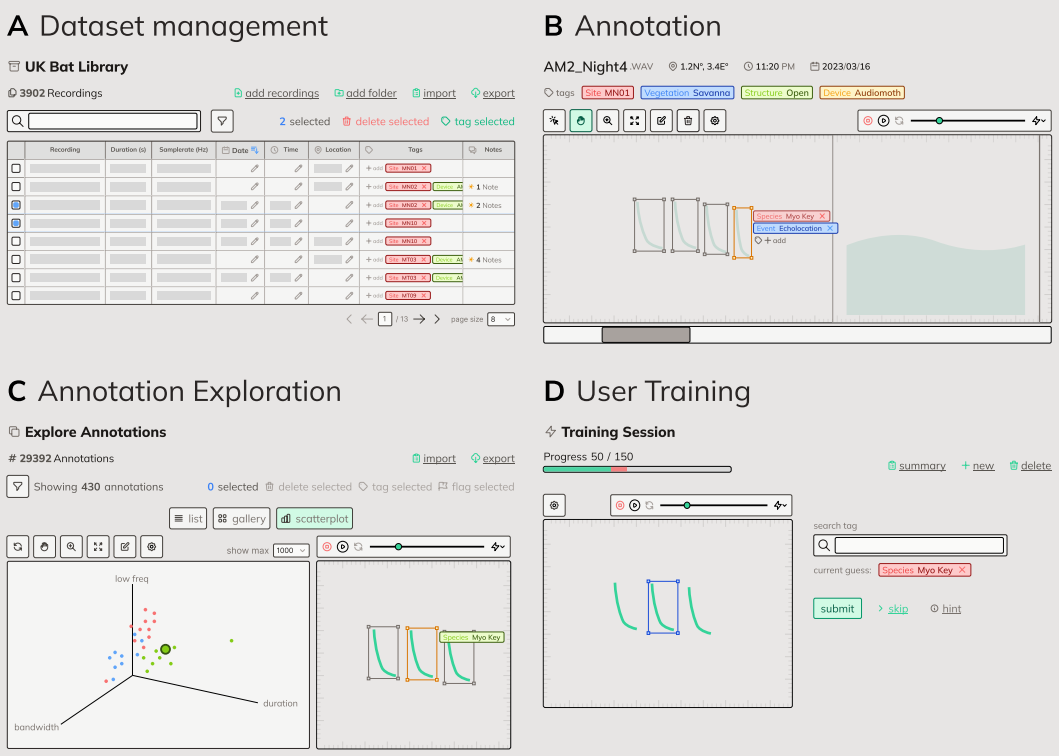


Figure 2.2: Overview of the key features of `whombat`. A presents dataset management capabilities, including editing, searching, and filtering recording metadata. **B** showcases the main annotation interface, offering spectrograms of recordings and various annotation tools. **C** demonstrates annotation exploration features, enabling users to browse existing annotations through filtering and visualisations, such as scatter plots. Finally, **D** highlights the training component of `whombat`, where users learn to identify a collection of sound events while receiving guidance on sound event identification. The panels shown are a design mock-up of the user interface, with some elements removed for clarity.

in their entirety. Users can zoom in to relevant parts of the spectrogram or zoom out to scan for interesting sounds. `whombat` also provides searching, filtering, and sorting tools to quickly browse the recordings of interest.

2.3.2 Annotation

Annotation projects can be created by selecting any number of audio clips from recordings of interest. Audio clips are continuous sections extracted from recordings. They can vary in duration and are not constrained to match the length of the original recording. The use of audio clips as the basis of annotation tasks allows cutting the recordings into clips of standardised duration and possibly annotating only a subset of all audio clips. The included clips can be selected within the tool or imported

from a CSV file. To create an annotation project, a name, description, and annotation instructions for the annotators should be provided.

Once an annotation project is created, each audio clip can then be visualised and annotated. A configurable spectrogram of the clip is displayed, along with recording metadata to provide context to the annotator (Fig. 2.3). Annotation can proceed in different ways depending on the project targets and strategy. Users can add any number of key-value tags to the recording clip, for example to specify which species are present within the clip. Relevant sound events can be annotated by locating them within the spectrogram by drawing a vertical line, a temporal interval or a bounding box. Each annotation can be tagged with any amount of key-value pairs, potentially capturing multiple and independent attributes of the sound event, such as species, sound type, sex, or the identity of the individual. Although tags can be created freely, `whombat` offers a quick search feature to avoid duplication and to ensure consistency.

As annotations progresses, audio clips can be marked as "ready" once they have been fully annotated according to the project instructions. Annotation progress is tracked by displaying the percentage of audio clips that have been marked as ready, along with the counts of annotated clips and annotations with a given tag. With the aid of filtering and sorting tools, users can focus and prioritise their annotation efforts on specific subsets of the annotation project.

2.3.3 Review and exploration

Quality of metadata and annotations can be reviewed and managed through various tools within `whombat`. Users can add notes to recordings and annotations to provide additional context and note issues that require fixing. Incomplete annotations can also be flagged, and the issues can be searched to address them efficiently.

In addition, `whombat` provides tools for exploring and comparing groups of annotations. The gallery option displays a panel of annotated sound events from different user-selected groups, allowing for easy comparison. For groups of bounding box annotations, the tool can compute statistics on attributes such as the duration, bandwidth, and frequency range, and display them in histograms. `whombat` also

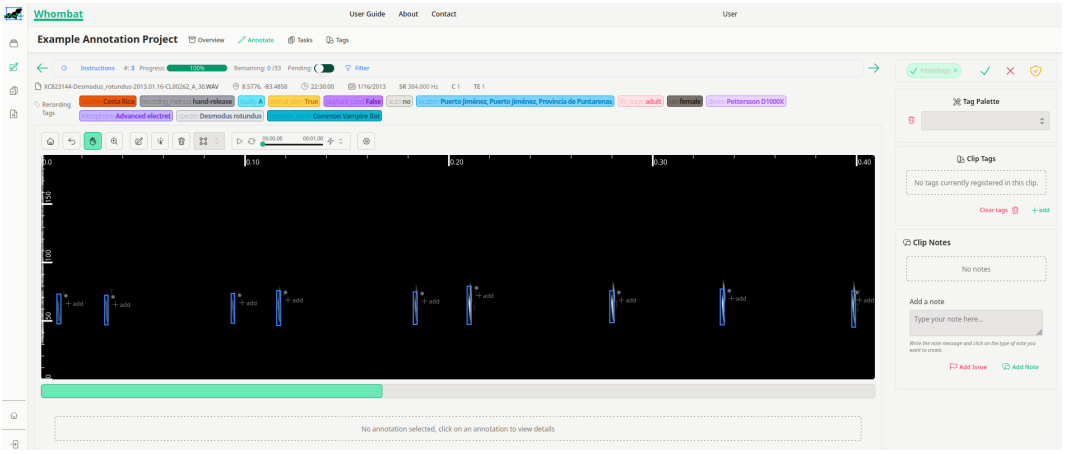


Figure 2.3: The `whombat` audio annotation interface. The interface presents a spectrogram visualisation of the audio clip (center) along with tools for task navigation (top bar), clip-level tagging and notes (right sidebar), and detailed sound event annotation creation and editing (within the spectrogram).

provides an interactive 2D or 3D scatter plot of any combination of the previously mentioned attributes (Figure 2.2). These visualisation tools enable users to become familiar with the variety of sound events and identify potential issues such as outliers and overlaps between categories.

2.3.4 User training

Novice users can be incorporated into the workflow by training with existing verified data, which is critical as access to experts in bioacoustics is a recurrent bottleneck for annotation. Our tool addresses this issue by allowing users to learn and improve their annotation skills. The registered annotations can be used to train and evaluate human annotation skills. Users can create training sets by selecting specific annotations, such as those with a particular set of tags. The training sets can be used to conduct training sessions (Figure 2.2), in which users are shown a series of spectrograms centred at annotated sound events and asked to identify them correctly. After each session, the identification performance is evaluated and displayed, enabling users to track their learning progress and identify areas that need improvement.

2.3.5 Data export

`whombat` allows users to export their acoustic datasets and annotation projects to multiple file formats. The recommended format is a custom JSON format inspired by

the COCO dataset format (Lin et al., 2014), although a CSV format is also available. This makes it possible to use the annotations for training DL identification models, other bioacoustic analysis, or to share with the wider community.

In addition, the exported dataset and annotation project files can be imported back into the tool. This functionality allows for offline distributed collaborative work where multiple people work on disjointed datasets and share the resulting annotations. This is particularly useful because it bypasses the need for centralised server infrastructure.

2.3.6 Closing the loop

To improve DL model performance, the tool provides a way to import model predictions and compare them with user-made annotations. `whombat` accepts model predictions in a specific JSON or CSV format, with no restriction on the type of DL model used. Once imported, the set of predictions for a group of recordings is registered as a model run. Users can provide a name and description for the run to help track and organise different model experiments.

`whombat` then allows users to evaluate the model run by comparing it with annotations, if available. Several measures of predictive capacity, such as precision and recall, are displayed to help users assess the performance of the model. Users can also explore the predictions using search, sort, and filter tools, based on the predicted tag probabilities. This facilitates browsing both success and failure cases, helping to identify potential model improvement opportunities.

In addition to evaluating the DL model, the tool can also be used to diagnose potential data and annotation gaps. By comparing the model predictions with user-made annotations, users can identify cases where the model fails to detect sound events correctly. These cases can then be reviewed to see if there are annotation or data gaps that need to be addressed to improve model performance.

2.4 Use Cases

`whombat` is designed specifically for audio data in the field of bioacoustics, and its flexibility makes it adaptable to a range of use cases. In this section, I highlight two examples of how the tool can be used: annotation of bat calls from the UK and bird vocalisation detection in the Pacific Northwest of the USA. These examples showcase the versatility and potential of the tool for annotating different types of target species and vocalisations.

2.4.1 Bat call classification pipeline

The Bat Conservation Trust (BCT) uses `whombat` to improve bat detection and classification in the UK. The BCT collects and annotates recordings of bat calls across the UK to enhance the BatDetect tool (Mac Aodha et al., 2018) and advance from bat call detection to a multi-class object detection and classification pipeline (Chapter 4). Bats play a crucial role in the UK ecosystems (Barlow et al., 2015), and as small, nocturnal, volant mammals that use ultrasonic echolocation for navigation they are routinely monitored using passive acoustic methods (Banner et al., 2018; Barlow et al., 2015; Kerbiriou et al., 2015; Newson et al., 2015; Yoh et al., 2023). Furthermore, interspecific differences in bat echolocation call characteristics enables species or genus level identification from acoustic data. Automating the classification of bat echolocation calls enables monitoring to be carried out at the scales necessary for identifying national conservation management strategies. Improving the detection and classification performance of automated tools, such as BatDetect (Mac Aodha et al., 2018), is therefore crucial for the success of conservation efforts.

`whombat` has enabled the BCT to generate precise bat call annotations while offering flexibility in the types of annotations captured. Bat calls are short and high-frequency, making them well-suited for annotation with bounding boxes tightly placed around the main harmonic. Annotators use tags in the form `species:` <*species*> to indicate the bat species, and `event:` <*call type*> to specify the call type (e.g., echolocation, social call, feeding buzz). In cases of uncertainty, a generic tag like `order:` *Chiroptera* can be employed. Additionally, potential false positives

can be annotated with an `event`: *Noise* tag to reduce confusion. While bats are the main focus at the BCT, and it is important to be able to capture their different types of calls, it is also crucial to identify confounding noises and register co-occurring sounds that can be important for downstream analysis.

The tool has enabled the BCT to centralise annotation work, eliminating the complexities of harmonising previously independent efforts. Furthermore, `whombat` has allowed to streamline the review process by allowing to assess the annotator's work. This collaborative approach has proven valuable, leading to the identification and correction of mislabelled annotations due to confusion in the annotation instructions. This has allowed the BCT to improve both the quality and quantity of their annotations The collaborative nature of the tool also allows for efficient data sharing and analysis, making it an essential tool for BCT and their bat conservation work.

A total of 29 independent datasets of bat recordings, comprising over 70,000 annotated calls, have been processed at the BCT using `whombat`. It has been used by more than 15 independent annotators from the BCT and partner institutions. The annotations generated using `whombat` directly inform the training of DL algorithms (Chapter 4), demonstrating improved performance compared to other existing bat detection tools. These annotations and the refined models they enable extend the BCT's capability to understand bat population responses to anthropogenic environmental change and inform conservation efforts.

2.4.2 Bird song annotation

In 1994, the Northwest Forest Plan was introduced in the United States Pacific Northwest to shift federal land management policies from prioritising timber harvesting to a more holistic approach that includes protecting and restoring the habitat of old-forest species and biodiversity (Espy & Babbitt, 1994). One of the components of this plan is the long-term monitoring of federally threatened northern spotted owl (*Strix occidentalis caurina*) populations through a two-phase approach (Lint, 1999). The first phase involved estimating vital rates and demographic performance using mark-resight methods on historical territories (Franklin et al., 2021). The

second phase began in 2020 and focused on estimating occupancy and habitat models through passive acoustic monitoring (Lesmeister & Jenkins, 2022).

The transition to phase two monitoring is a crucial moment in conserving and managing forested lands in the Pacific Northwest. Not only are spotted owl conservation and management objectives being met (Lesmeister & Jenkins, 2022; Weldy et al., 2023), but the multispecies acoustic monitoring data can also be used to address other conservation, research, or management objectives. To this end, researchers from Oregon State University and the USDA Forest Service are using `whombat` to annotate avian sounds (> 30,000 annotations) and validate model predictions for various wildlife monitoring programs targeting federally threatened species like the northern spotted owl and the marbled murrelet (*Brachyramphus marmoratus*), as well as sensitive species such as the white-headed woodpecker (*Dryobates albolarvatus*), and supporting broader biodiversity monitoring efforts (> 80 species).

The dynamic acoustic and spectrogram adjustments provided by the tool have improved the quality of target species annotation, increased efficiency in reviewing model predictions, and aided in tracking acoustic review and labelling efforts. In addition, the annotation formatting of `whombat` is flexible and dynamic, allowing annotators to pursue multiple annotation objectives simultaneously. They can opportunistically collect biophonic examples for non-target species and create hierarchical label structures where sound types are nested within broader categories. These hierarchical labels cascade across increasingly fine-scale taxonomic determinations. Additionally, annotators can label acoustic metadata such as approximate distances or overlapping sound types, which serves to improve model training and enhance the understanding of model performance. The adoption of passive acoustic monitoring represents an important step forward in conserving and managing forested lands in the Pacific Northwest. Using innovative tools such as `whombat` enhances these efforts.

2.5 Discussion

The modularity and extensibility of `whombat` enables many opportunities for future development (see the Appendix A.2 for more details on the software design). I invite the community to contribute to its growth and suggest potential areas of improvement, such as the ability to group annotations into sequences, model comparison and data iteration visualisations, and dashboards for ecological insights and quick exploration. One possibility for expanding the user base of the tool is to incorporate a citizen science approach by evaluating the reliability of user annotations. I believe these and other potential directions will help make `whombat` an even more powerful tool for bioacoustic research and conservation efforts.

Unlike other annotation solutions (e.g. Marsland et al., 2019), this tool does not include embedded Deep Learning detectors. I made this decision to simplify the software and decouple the annotation process from the development and maintenance of DL models. Instead, our focus is on providing a user-friendly interface for efficient and accurate annotation. I also provide an interface for importing and exporting model predictions, allowing users to incorporate their own DL models into their annotation projects. Additionally, the tool allows exporting annotations in a format that is compatible with training frameworks for bioacoustic detection models (Chapter 4).

By providing an accessible, open-source tool for bioacoustic annotation, I hope to empower research teams to generate high-quality acoustic datasets for their projects, including those without extensive coding experience. The modular and extensible design of the software allows for customisation to meet individual project needs and encourages community involvement in the development of new features. By lowering the barrier to entry for annotation projects, I aim to foster the creation of diverse and shareable datasets that can advance research in bioacoustics.

Chapter 3

Detailed Annotations Boost Classification Performance in Automated Acoustic Identification of Bats

3.1 Abstract

Acquiring training data for Deep Learning (DL) models in automated species detection is challenging, requiring efficient use of existing bioacoustic data. While manual annotation can enhance training data and improve model performance, there is no consensus on the most effective annotation method. Utilising a dataset of bat call recordings with expert-derived annotations, I investigated how detail of call location within the recordings affected the performance of DL models in bat call detection and classification calls from 17 species from Mexico across a range of dataset sizes (from 5 to 25 calls). I first established a baseline by training a Convolutional Neural Network (CNN) model using clip annotations, which simply indicate whether a call is present within an audio clip. Then, I trained models to additionally predict call location in time and frequency with varying levels of detail—using (1) only onset; (2) onset and offset; (3) onset, offset, and frequency bounds (defining a bounding

box in the spectrogram); and (4) the full time-frequency trajectory (a line-string representation of the call's frequency modulation)—evaluating their performance on a test set of unseen recordings. I found that employing detailed annotations consistently improved classification performance across all dataset sizes. The most significant gains were observed for datasets containing 10–20 recordings per species, ranging from 5% to 10% improvement. These performance gains from detailed annotations were comparable to, or exceeded, those obtained from increasing the dataset size by 5 recordings per species. I found no consistent and statistically significant differences in classification performance between the different levels of detail. This study demonstrates that annotating call location in time and frequency is a valuable strategy for enhancing the performance of deep learning models in bat call detection and classification, particularly given the substantial difficulties associated with collecting new reference recordings. I recommend using bounding boxes to annotate call location, as they offer a practical balance between annotation effort and model performance. This detailed annotation approach has the potential to significantly improve the efficiency of bioacoustic data utilisation for training DL models, highlighting the value of investing in thorough annotation.

3.2 Introduction

The field of acoustic biodiversity monitoring is increasingly leveraging Deep Learning (DL) for various tasks, including automated species detection and classification (Stowell, 2022; Tuia et al., 2022). Emerging technologies for automated acoustic identification of birds (Kahl et al., 2021), bats (Mac Aodha et al., 2018), and other soniferous taxa (Allen et al., 2021; LeBien et al., 2020) allow the study of the activity patterns of these species at large scales (Sethi et al., 2024). Training DL models for acoustic detection typically relies on the availability of sufficient reference recordings of the target vocalisations (Kaplan et al., 2020; Kahl et al., 2021), and alternative approaches that could reduce this reliance tend to be less effective when discriminating between classes with subtle distinctions (Cole et al., 2022), as is often the case in bioacoustics. However, acquiring new reference recordings, especially

in natural environments, can be expensive and logistically challenging (Pichler & Hartig, 2023). This inherent limitation in data collection creates a gap in the ability to train effective DL models for data-scarce regions. Therefore, investigating methods that improve the utility of existing but scarce bioacoustic data for training DL models is crucial for expanding the application of these models to a wider range of species and regions, ultimately enabling their automated monitoring at scale.

Training and evaluating DL models for acoustic identification requires providing the model with numerous examples of audio clips containing the target species (Stowell, 2022). These examples typically come from longer recordings where the target species has been confirmed, often requiring expert analysis, field observation, or even capturing the specimen to ensure accurate identification (Oswald et al., 2022; Gibb et al., 2018). Because many common DL classification and detection models operate on short, fixed-length audio clips (Kahl et al., 2021; Hershey et al., 2017), these longer recordings must be divided into smaller clips suitable for model training. Each clip needs to be accurately labelled as either containing the target vocalisation or not; however, without explicit manual labelling, this can prove challenging (Kong et al., 2019). Some studies address this by assuming all clips from a recording contain the target species or by using simple methods to detect prominent sounds (Chen et al., 2020; Kobayashi et al., 2021; Kahl et al., 2021; Bermant et al., 2019), which are then assumed to be the target species. However, these approaches can be inaccurate in environments with background noise, overlapping vocalisations from multiple species, and long silent periods between calls.

Manual annotation, pinpointing the time and/or frequency location of each target sound event within a recording, can provide DL models with more focused training data. This involves expert identification and localisation of each sound event, enabling more accurate selection of audio clips that contain (or do not contain) the target sounds. Various methods exist for annotating the location of a sound event, each with a different level of detail in capturing its location within recordings (Figure 3.1). The simplest approach, ‘clip annotation’, involves confirming the presence or absence of the target sound events within a fixed-duration audio

clip (Khalighifar et al., 2022; see Figure 3.1B). More detailed temporal annotation of each sound event is most commonly done by marking both the onset and duration (Morfi et al., 2019; Cañas et al., 2023), although sometimes only the onset time is marked for short, transient sounds (Lostanlen et al., 2018; see Figure 3.1C-D). If the target sound event covers a clear and bounded frequency range, the lowest and highest frequencies are also annotated, creating a 'bounding box'around the sound event in the spectrogram ((Hagiwara et al., 2023); see Figure 3.1E). When animal vocalisations have a distinct peak-frequency, that peak can be tracked through time using a 'line-string'annotation (Figure 3.1F) that follows the vocalisation's changing frequency. This approach, though not yet widely explored, could provide a highly detailed and informative form of annotation. While manual annotation is time-consuming, with the effort required varying by the level of detail, Hershey et al. (2021) found that using annotations solely to determine the presence or absence of target sound events in clips improved model performance. However, their study utilised a subset of AudioSet (Gemmeke et al., 2017), a large-scale dataset of diverse audio events not specific to bioacoustics, and did not leverage any additional details of the annotations, which may be crucial for discriminating between acoustically similar species.

Detailed annotations offer more than just indications of which clips contain target sound events. While the traditional approach uses only the clip label (i.e., whether the target species vocalises within the clip), this disregards valuable information about the vocalisation's location within the clip. In computer vision, tasks like object detection rely heavily on annotated location information (Lin et al., 2014). For example, bounding box annotations are used to train models that directly predict the location of objects within images (Zou et al., 2023), and a similar approach has been proposed for general audio tasks (Pham et al., 2018). Alternatively, the location information can be used in a multi-task learning approach (Stowell, 2022; Martin et al., 2022), where models are trained not only to identify the presence of the target sound event within a clip but also to predict its location within it. This approach, where a single model performs several tasks concurrently, has been shown

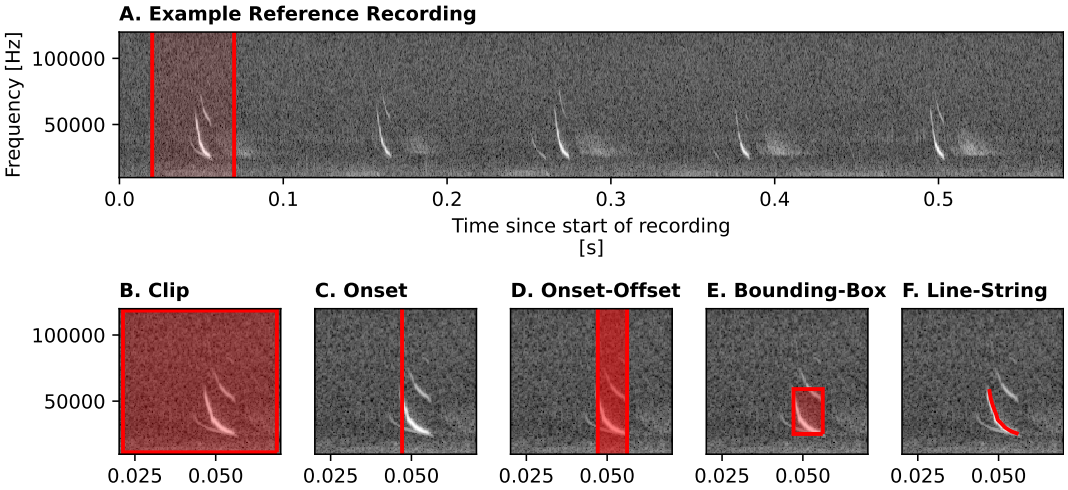


Figure 3.1: Common bioacoustic annotation types. (A) Spectrogram of a reference recording containing *Lasionycteris noctivagans* echolocation calls. These recordings are often lengthy and contain multiple vocalizations from the same individual, interspersed with silence and other non-target sounds. When developing deep learning (DL) models for the detection and classification of animal vocalizations, smaller clips of fixed duration (red segment) are used as training examples. However, without manual review, it can be difficult to determine if a target sound is present within a clip. (B–E) Examples of manual annotations of the target sound with increasing location detail (red highlights): (B) Clip annotation confirms presence within a shorter clip. (C) Onset annotation marks the start of the target sound event. (D) Onset-offset annotations mark the start and end time of the target sound event. (E) Bounding box annotation marks the lower and upper frequency bounds, as well as the onset and offset. (F) Line-string annotations trace the peak frequency throughout the target sound's duration.

to improve performance in various domains (Amyar et al., 2020), likely because learning characteristics useful for multiple tasks helps to avoid over-fitting (Zhang & Yang, 2022), a common issue when training with limited data. Therefore, given the variety of annotation methods and their potential uses, it is crucial to study the benefits of each method on model performance and understand how they depend on the amount of training data available, ultimately enabling the selection of the best approach for datasets with limited data.

The limitations of collecting new bioacoustic data are particularly evident when studying bats. Bats, which perform essential ecosystem services like pollination, seed dispersal, and insect population control (Jones et al., 2009), are readily detectable acoustically using well-established manual and semi-automated approaches (Zamora-Gutierrez et al., 2021). As one of the most studied taxonomic groups using terrestrial

passive acoustic monitoring techniques (Sugai et al., 2018), numerous efforts have focused on automating their detection and classification in passive recording (Mac Aodha et al., 2018; Kobayashi et al., 2021; Chen et al., 2020; Khalighifar et al., 2022; Yoh et al., 2022; Vogelbacher et al., 2023; Tabak et al., 2022). However, despite their significance, assessing bat population species trends for most species remains challenging, especially in the data-deficient regions like the tropics (Frick et al., 2019). This difficulty partly stems from the challenge of acquiring reference recordings. Unlike birds, which are often identifiable visually or by their distinctive songs, reliable ways to identify the recorded species often involves capture and release, a time-consuming and potentially stressful process for the animals (Zamora-Gutierrez et al., 2020). This underscores the importance of maximising the value of existing data through detailed annotation, which is facilitated by the clear temporal and spectral patterns of bat echolocation calls. With a growing number of public reference libraries (Görföl et al., 2022; Zamora-Gutierrez et al., 2020; Vellinga & Planque, 2015) and the expectation of more reference recordings being shared, consideration of how best to annotate these resources is pressing.

Here, I investigate how the different manual data annotation methods influence the performance of DL models, using bat detection and classification as a case study. To this end, I compile and annotate a dataset of Mexican bat call recordings and use it to develop a common detection and classification pipeline for 17 species incorporating a Convolutional Neural Network (CNN). I train several CNN variants using various annotation schemes (clip, onset, onset-offset, bounding box, and line-string, Figure 3.1) and across different training data sizes to assess the impact of annotation strategies on model performance. To leverage the information in the detailed annotations, I employ a multi-task learning approach where models simultaneously perform detection, classification, and location prediction of bat calls within audio clips. I hypothesise that by requiring the model to perform this additional location prediction task during training, the model will learn to prioritise features extracted from time-frequency regions of the spectrogram that contain bat calls. This, in turn, should improve learning efficiency and boost performance,

especially when training data is scarce and when using more precise annotations.

3.3 Materials and Methods

3.3.1 Acoustic data

I utilise bat call recordings from two primary sources. Firstly, recordings originated from the Sonozotz project (Zamora-Gutierrez et al., 2020), which employed a rigorous capture and release protocol. This protocol ensured coverage of diverse recording and release methods and utilised consistent high-quality recording equipment (Avisoft UltraSoundGate 116H@; Avisoft Bioacoustics). Captured individuals were identified by experienced Mexican bat researchers using a combination of morphometric measurements and visual inspection. The second source consists of bat call recordings donated to the Mexican Comission for the Knowledge and Use of Biodiversity (CONABIO). These recordings were all of known species, either captured individuals or recordings from known species roosts. Unlike the Sonozotz data, this donated collection exhibits greater heterogeneity in recording devices, settings, and processing methods, with 65% time expanded recordings. While predominantly captured in Mexico (79%), the remaining 21% originated from other countries. From these sources, I selected a total of 2,457 recordings (1,156 from Sonozotz and 1,301 from donated recordings) focusing on species confirmed to be present in Mexico. While the dataset encompasses reference recordings for the 101 bat species, it exhibits a significant class imbalance. Forty-seven species have fewer than 10 recordings each, while the most frequently recorded species, *Antrozous pallidus*, has 192 recordings. These recordings have an average duration of 2.2 seconds, with some lasting up to 11 seconds.

Bat experts reviewed and annotated each discernible bat echolocation call within the recordings. The location of the main harmonic of each call—the harmonic component containing the peak amplitude—was annotated using a line-string (Figure 3.1F). The main harmonic is commonly used for species identification and call characterisation (Szewczak, 2004), and using a line-string provided the highest level of detail possible, from which all other types of annotations (clip, onset,

onset-offset, bounding-box, see Figure 3.1B-E) could be derived. Species identification was attempted for each call, and since most calls likely originated from the recorded individual, species assignment was usually straightforward. When vocalisations from other species were present, annotators identified the species only if they were completely certain based on the visual characteristics of the call. Calls where acoustic identification remained uncertain were assigned the class *Chiroptera*. The resulting annotated dataset contains 51,461 annotated echolocation calls. A custom user interface developed at CONABIO was used in early stages of the annotation process, but review and finalisation of the annotations were performed using `whombat` (Chapter 2).

To develop and evaluate bat detection and classification models, I split the recordings into a test set and a development set used for both training and validation. To assess whether models generalise to novel recording conditions, I employed a geographic location-based split strategy across sampling locations in Mexico (see Appendix B.1 for details on the split). This split resulted in 740 recordings from 58 different locations being allocated to the test set, while the remaining 1717 recordings from 156 locations comprised the development set. From the initial 101 species, I selected a subset of 17 as the focus for the classification models, grouping all other species into the generic *Chiroptera* class. This selection criterion was necessary to ensure sufficient representation in the test set, as having fewer than 5 recordings per species could lead to unreliable performance evaluations. While each recording could have many echolocation calls, these calls are not truly independent because they likely originate from the same individual. To more accurately represent the number of independent observations, I used the number of recordings instead of the total number of calls. See the appendix for a detailed breakdown of the resulting dataset (Appendix B.1; Table B.1).

3.3.2 Detection and classification pipeline

I adopted a two-stage pipeline commonly used for developing automated bioacoustic detectors and classifiers to detect and classify bat calls within recordings. The first

stage involves preprocessing the audio by computing the spectrogram and segmenting it into fixed-duration clips. The second stage employs a Deep Learning (DL) model to detect the presence of target sounds within each clip and, if present, predict the species. This approach has been successfully applied to various bioacoustic tasks, including the analysis of both general bioacoustic signals (Kahl et al., 2021; Allen et al., 2021; LeBien et al., 2020; Ghani et al., 2023) and bat echolocation calls (Mac Aodha et al., 2018; Chen et al., 2020; Schwab et al., 2022).

For preprocessing, I followed procedures similar to those in previous bioacoustic work (Mac Aodha et al., 2018). I first resampled all recordings to 441 kHz, the most frequent sampling rate in the dataset, to standardise the sampling rates. Then, I applied a Short Time Fourier Transform (STFT) using a window length of 512 samples, 75% overlap, and a Hann window. Although most recordings targeted individual bats with high-quality recording equipment, they were primarily collected in the field. To reduce environmental noise and highlight bat calls, I employed the Per-Channel Energy Normalisation (PCEN) transformation (Wang et al., 2017). I segmented the spectrograms into 50ms clips with a 25ms overlap using a sliding window. This clip duration ensures that the echolocation calls of all target bat species, including those with longer calls (e.g., 35ms calls by *Pteronotus parnellii*), are not segmented. Finally, each spectrogram segment was resized to a 128x128 array and normalised.

For the second stage, I used a Convolutional Neural Network (CNN) for bat call detection and classification within each 50 ms clip. The model employs an encoder architecture comprising several convolutional layers followed by max-pooling operations. These layers reduce the clip's spectrogram into a compact set of features, which are then fed into two separate heads: a detection head and a classification head (Figure 3.2A). The detection head predicts whether the clip contains a bat echolocation call, regardless of the species, including both target and non-target species. The classification head classifies the clip as one of the target species or the generic *Chiroptera* class. This dual-head approach, similar to the method described by Schwab et al. (2022), allows the model to perform both

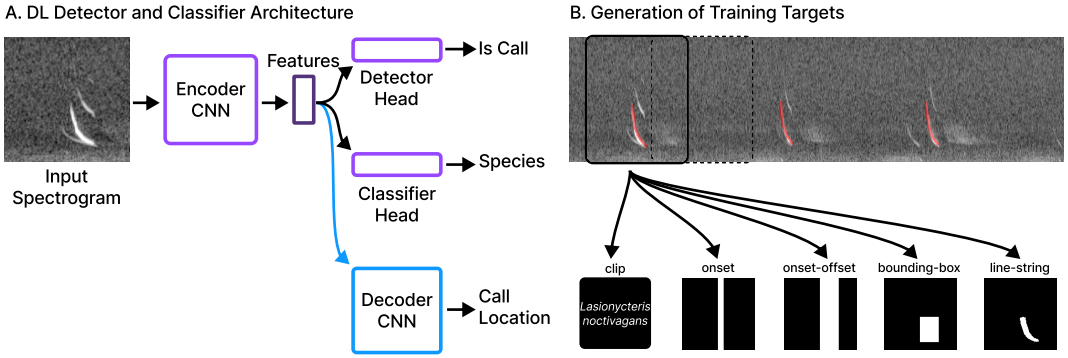


Figure 3.2: Detector and classifier architecture and training targets. (A) Model architecture for bat call detection and classification. The spectrogram of a fixed-duration audio clip is fed into a CNN encoder that extracts relevant features. These features are then passed to a detector head and a classifier head. The detector head identifies whether a bat call is present in the input spectrogram, while the classifier head predicts the species. Optionally, the features can be fed to a decoder CNN (blue) that predicts the locations of bat calls within the spectrogram. (B) Generation of training targets from manual annotations. The recording's spectrogram is segmented into fixed-duration clips using a sliding window; the black box represents a single clip, and the dotted black box represents the next clip in the sequence. The detailed line-string annotations (red) are used to generate clip labels (clip), indicating the presence and species of a bat call within each clip. Additionally, the annotations can be used to generate binary masks (onset, onset-offset, bounding box, line-string) that provide the location of the calls within the spectrogram with varying levels of detail. The CNN models are trained using the clip labels as targets for detection and classification tasks, or with an additional localization task using one of the binary masks.

detection and classification tasks within a single forward pass. I trained this CNN model using various strategies to incorporate different levels of annotation detail, as described in the following section.

For model training, I used consistent settings across all model variants. Specifically, I used a batch size of 32 and trained for a maximum of 100 epochs with early stopping to prevent overfitting. I monitored the classification balanced accuracy (ACC) on the validation set after each epoch and stopped training if this metric did not improve for more than 3 consecutive epochs. I used the Adam optimizer with a learning rate of 0.0001 and cosine annealing. To maintain consistency with established approaches, I adopted the hyperparameters used by Mac Aodha et al., 2018. All models were trained using PyTorch (Paszke et al., 2019) and PyTorch-Lightning (Falcon & The PyTorch Lightning team, 2019) in a "p3.2xlarge" instance at Amazon Web Services with a single Tesla V100 GPU. Training runs took between 10 and 40

minutes, with the duration depending on both dataset size (5 or 25 recordings per species) and model variant. Detailed model variants required approximately twice the training time of the clip variant. See the Appendix B.2 for further details on the model architecture.

3.3.3 Evaluating the impact of location detail on model performance

As a baseline, I trained a CNN model (CNN_{clip} model) using only clip-level labels to simulate a scenario where detailed annotations are unavailable. This approach mirrors Hershey et al. (2021), where annotations were used solely to determine the presence or absence of target sound events. I labelled a clip as positive for a bat echolocation call if it overlapped with any line-string annotation. If the overlapping annotation included a species label, that label was assigned to the clip; otherwise, a generic "Chiroptera" label was used. For clips containing multiple annotated calls, I prioritised the species label of the first call. All other clips were labelled as "empty," indicating the absence of a detectable call. These clip-level labels served as targets for both the detection and classification heads of the CNN model (Figure 3.2B).

To investigate whether incorporating detailed time-frequency location information could improve model performance, I trained model variants with an additional localisation task. These variants, referred to as CNN_{onset}, $CNN_{onset\text{-}offset}$, $CNN_{bounding\text{-}box}$, and $CNN_{line\text{-}string}$, simulate scenarios where annotations of the corresponding type are available during training. These models were trained to perform three tasks simultaneously: detecting bat calls, classifying their species, and predicting their location within the input spectrogram. To enable localisation, I augmented the CNN architecture with a decoder component. This decoder uses the features extracted by the encoder to predict a "location mask" highlighting the pixels in the input spectrogram where bat calls occur. Specifically, the decoder uses a series of transposed convolutional layers to upsample the encoder features, followed by a final classification layer that predicts, for each pixel in the upsampled mask, whether it belongs to a bat call. During training, the input spectrogram is passed through the

encoder to generate features used for detection, classification, and localisation (Figure 3.2A). The model is trained to minimise the combined loss from the detection, classification, and localisation tasks. However, during testing, only the detection and classification outputs are used to ensure a fair comparison with models trained without location information. This multi-task training encourages the encoder to learn a feature representation that is sensitive to both the acoustic characteristics of different bat species and the precise location of calls within the spectrogram.

To generate the localisation training targets for these variants, I first converted the original line-string annotations to the other annotation types, as needed. For instance, to create the onset-offset annotation, I determined the bounding box of the line-string and retained only the onset and offset points. Each annotation was then rasterised to create a binary mask with the same dimensions as the input spectrogram. During this rasterisation process, I incorporated a small buffer around each annotation to account for potential inaccuracies. Specifically, a pixel was considered to belong to a bat call if it fell within 2ms and 2kHz of the corresponding annotation. For example, a pixel at time 1s and 60kHz would be considered part of an onset-offset annotation starting at 0.9s and ending at 1.1s, but not part of a bounding box annotation with the same temporal bounds but with frequency bounds of 10-20kHz. Each pixel in the mask was assigned a value of 1 if it belonged to a bat call (regardless of species) and 0 otherwise. During training, each input spectrogram clip was paired with its corresponding annotation mask, which served as the target for the localisation task. The labels for the detection and classification tasks were generated in the same way as for the clip model variant. Full details on the specific loss functions used for each model variant can be found in the Appendix B.3.

3.3.4 Model evaluation

To evaluate the performance of the CNN models, I used two metrics that capture both detection and classification accuracy. Each trained model was applied to the entire test dataset using the sliding window approach. Notably, this includes clips that contain no bat calls, which are essential for evaluating detection performance.

For each clip, I recorded the confidence score of the model for detection and the scores for each target species and the generic Chiroptera class. Ground truth labels were derived in the same way as the training labels, by identifying annotations that overlapped with the clip. I used average precision (AP) for detection, as it provides a unified measure of performance across different confidence score thresholds. For classification, I used balanced accuracy (ACC) to account for the varying number of examples per species in the test set. Balanced accuracy was evaluated only on clips containing a bat call.

To evaluate the influence of training data size on model performance, I trained each model variant on nested datasets of increasing sizes (5, 10, 15, 20, and 25 recordings per species). For each of the 17 target species, I randomly selected 5 recordings for validation, using the remaining recordings to create nested training sets. To control for variability arising from data partitioning, I performed five independent training runs per dataset size, each using a different random split into training and validation sets. Each run utilised a different random split of the corresponding data into training and validation sets. Within each run, all model variants were trained on the same data with identical encoder initialisation, enabling a fair comparison of their performance across data partitions and dataset sizes.

To assess whether differences in performance between model variants were statistically significant, I performed paired t-tests on variants trained on the same datasets (Raschka, 2020). I considered differences to be significant at a p-value threshold of 0.05.

3.4 Results

3.4.1 Impact of annotation detail on classification

The CNN model variants trained with detailed annotations (CNN_{onset}, $CNN_{onset\text{-}offset}$, $CNN_{bounding\text{-}box}$ and $CNN_{line\text{-}string}$) significantly improved classification performance across most dataset sizes compared to the CNN_{clip} baseline (Figure 3.3). This improvement was particularly pronounced with 10 and 15 recordings per species,

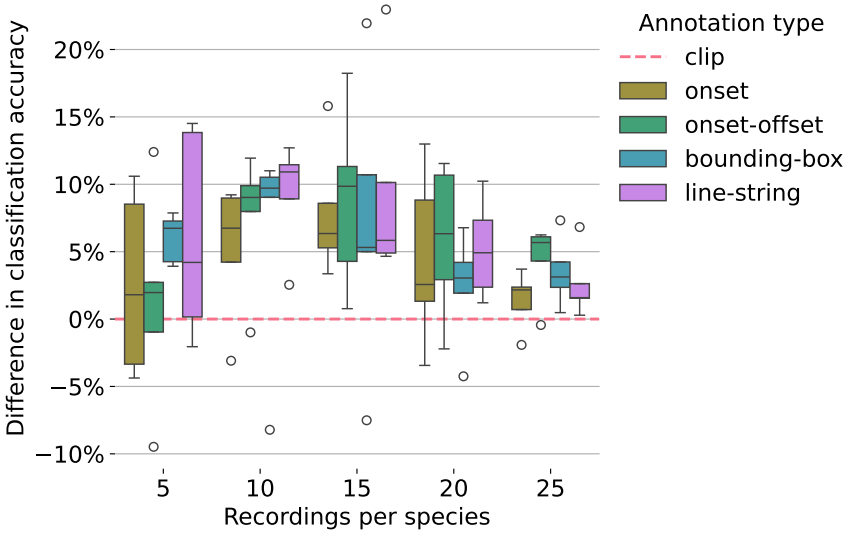


Figure 3.3: Comparison of the classification performance against the CNN_{clip} baseline. Boxes show the distribution of the difference in classification performance measured by balanced accuracy between the detailed variants and the CNN_{clip} variant. The red dashed line represents no change in performance. Models variants were trained 5 times for each dataset size and evaluated in a held-out test set.

where classification accuracy increased by 5-10%. Even with 25 recordings per species, a significant 2-5% gain in accuracy was observed. However, with only 5 recordings per species, the impact of detailed annotations was less consistent and less pronounced.

Increasing the dataset size improved classification performance, regardless of the annotation type (Table 3.1). However, performance gains were more pronounced for smaller datasets with detailed annotations. Notably, using detailed annotations with a smaller dataset could achieve similar performance improvements to increasing the dataset size by 5 recordings per species (Table 3.1). Classification performance showed diminishing returns with 25 or more recordings per species, but the absence of a clear plateau in the performance curve suggests that larger training datasets might yield further improvements.

Despite differences in mean classification accuracy, there was no consistent and statistically significant differences in classification performance between the detailed annotation variants (CNN_{onset}, $CNN_{onset\text{-}offset}$, $CNN_{bounding\text{-}box}$, and $CNN_{line\text{-}string}$) across various training scenarios. However, I observed some trends when examining

Table 3.1: Classification performance of model variants. Balanced accuracy (ACC) scores are shown for each model variant trained using its corresponding annotation type, across different training dataset sizes (number of recordings per species). Results are averaged over 5 training runs, with each run using identical training data, hyperparameters, and initialization weights for all model variants. Within each dataset size, the model variant with the highest average ACC is shown in bold. The superscript denotes the number of times (0 if absent) that the variant achieved the highest ACC for that dataset size.

	Balanced accuracy (%)				
Recs/species	CNN_{clip}	CNN_{onset}	$CNN_{onset\text{-}offset}$	$CNN_{bounding\text{-}box}$	$CNN_{line\text{-}string}$
5	30.3	32.9	31.6	36.3^{2}	$\mathbf{36.4}^{3}$
10	39.1	44.3	46.7	45.5^{1}	$\mathbf{48.4}^{4}$
15	43.1	50.9^{1}	52.0^{1}	50.2	$\mathbf{52.8}^{3}$
20	51.5	55.9^{1}	$\mathbf{57.3}^{3}$	53.8	56.7^{1}
25	55.7	57.1	$\mathbf{60.1}^{2}$	59.2^{2}	58.3^{1}

the frequency of top-performing models (Table 3.1). The $CNN_{line\text{-}string}$ variant, which provides the most detailed annotation, produced the most top-performing classifiers (12 out of 25 training configurations). This trend diminished with larger datasets, where $CNN_{bounding\text{-}box}$ and $CNN_{onset\text{-}offset}$ variants also yielded top-performing models.

3.4.2 Impact of annotation detail on detection

In contrast to the classification results, detection performance showed little variation across dataset sizes and annotation variants (Table 3.1). Average precision remained consistently high ($\sim 95\%$), with no notable differences observed between models trained on different dataset sizes or with different annotation types. The baseline model, trained with clip annotations and only 5 recordings per species, achieved an average detection performance of 94.5% in average precision.

3.5 Discussion

In this study, I show that providing detailed annotations of the time-frequency location of calls within the spectrogram can improve the performance of Deep Learning (DL) models for bat echolocation call detection and classification. The results show a consistent increase in classification performance across all dataset sizes when using detailed annotations, with the most significant gains observed for

Table 3.2: Detection performance of model variants. Average precision (AP) scores are shown for each model variant trained using its corresponding annotation type, across different training dataset sizes (number of recordings per species). Results are averaged over 5 runs. Within each dataset size, the model variant with the highest AP is shown in bold.

	Average precision				
Recs/species	CNN_{clip}	CNN_{onset}	$CNN_{onset\text{-}offset}$	$CNN_{bounding\text{-}box}$	$CNN_{line\text{-}string}$
5	94.5%	94.3%	**94.5%**	94.2%	94.3%
10	94.6%	94.8%	94.5%	94.8%	**95.3%**
15	94.9%	94.0%	**94.9%**	93.9%	94.2%
20	**94.8%**	94.3%	94.4%	93.6%	94.4%
25	94.5%	94.5%	93.8%	93.6%	**94.8%**

smaller datasets (10–20 recordings per species). Although acquiring more data is generally recommended when possible, this finding suggests that detailed annotations are particularly valuable when training data is limited and collecting additional data is challenging. However, creating detailed annotations can be time-consuming. I measured annotation speed for a set of 20 one-second recordings. Clip annotations, which involve only a single action of assigning the appropriate species label (if any) to each clip, took approximately 3 minutes to complete. In contrast, onset, onset-offset, and bounding box annotations, requiring one or two interactions per call, took around 20 minutes, while line-string annotations, requiring multiple interactions per call, took roughly 40 minutes. These measurements provide a general approximation, as actual annotation time is highly dependent on event density, the specific user interface employed, and operator proficiency. A more robust evaluation involving multiple annotators and diverse target events would be valuable.

Based on the findings of this study (Fig. 3.3), we recommend that future annotation efforts, especially when working with limited data (e.g., fewer than 25 recordings per species), prioritise detailed annotation methods over simple clip-level annotations. Bounding box annotations offer a particularly effective balance between annotation effort and performance gains. While no single detailed annotation method consistently outperformed the others, bounding box annotations yielded statistically significant accuracy gains over the baseline clip-level model across all dataset sizes. This approach offers a practical balance between improved performance and annota-

tion effort, providing a high level of detail surpassed only by the substantially more time-consuming line-string annotations, which did not yield further performance gains.

My study revealed that the primary performance gain from incorporating detailed location information was observed in the classification system. This finding is likely related to the nature of bat echolocation calls, which can exhibit subtle inter-species variations in frequency ranges and call structures. Even though this study focused on only 17 bat species, the complete dataset encompasses a wider set of species with diverse call types, including some groups that are challenging to discriminate acoustically (Zamora-Gutierrez et al., 2016). The additional localisation task benefited model training and classification performance by providing more precise information about the relevant acoustic features within each call, enabling the model to learn finer-grained distinctions between species. Furthermore, since the recordings used were captured in the open environment, they contain background noise that could confound the models and contribute to overfitting. While preprocessing techniques like PCEN are widely used for noise reduction, they may not eliminate all non-target sounds. Guiding the training process by explicitly highlighting the location of the relevant signal through the localisation task could further mitigate the impact of noise and improve generalisation. These challenges are common in bioacoustic tasks, suggesting that this findings may extend to other datasets and taxa. However, further validation is needed to confirm the generalisability of these results.

In this study, I adopted a widely used classification and detection pipeline that relies on training a DL model to analyse fixed-duration audio clips. I chose a common CNN architecture to facilitate direct comparison with prior studies. However, the specific architecture can influence performance, and the field is constantly experimenting with alternative and innovative architectures. For example, the larger ResNet architecture used in BirdNet (Kahl et al., 2021) and the EfficientNet backbone employed in Perch (Ghani et al., 2023). Recently, transformer-based architectures have shown promising results in bat detection and classification (Vogelbacher et al., 2023). These alternative architectures tend to have a larger number of parameters, typically

requiring more data for effective training. Similarly, I adopted Mac Aodha et al., 2018 training hyperparameters for consistency, however, a dedicated hyperparameter optimisation process could potentially yield further performance improvements. Still, while architectural and hyperparameter refinements may lead to performance gains, I focused on investigating whether improvements in annotation effort could also enhance performance. My findings suggest that such improvements can indeed lead to better results, and I believe these results are likely transferable to other architectures, though further validation is needed.

It is important to note that the clip-based approach has inherent limitations in temporal resolution. With clip-level predictions, increasing temporal resolution requires processing more overlapping clips, thus increasing computational costs, or reducing clip duration, which limits the temporal context available for inference. Alternative approaches, such as the object detection method employed in Chapter 4, directly predict call locations, bypassing clip classification entirely and potentially offering higher temporal resolution. This fine-grained detection capability is valuable for downstream tasks like identifying feeding buzzes. These considerations raise important questions regarding the suitability of the clip-based approach for all bioacoustic tasks and motivate the exploration of alternative detection methods.

Another interesting finding is the consistently high detection performance across all model variants and training dataset sizes. This result demonstrates the feasibility of achieving effective, nationwide bat echolocation detection using a relatively small dataset. While the effectiveness of deep learning methods for bat call detection has been established (Mac Aodha et al., 2018), their performance in highly diverse settings with limited data remained an open question. In this work, models trained with only 5 recordings per species for 17 species achieved good detection performance (95% average precision) on a challenging test set encompassing recordings from across Mexico and containing 69 distinct species. Given that many bioacoustic datasets contain similarly low numbers of recordings per species (Nolasco, Singh et al., 2023; Chasmai et al., 2024), these results are encouraging, demonstrating the potential for effective detection of coherent taxonomic groups, like bats, even with

limited training data per species. It is important to acknowledge that test dataset used comprises focal recordings designed to capture echolocation calls from target individuals, which may differ substantially from those obtained through passive monitoring (van Merriënboer et al., 2024). While confirming species identification in echolocation calls from passive recordings can be challenging, creating synthetic test datasets (Salamon et al., 2017) that combine recordings of known species with passive recordings could help gain insights into detector performance under such conditions. Additionally, since the test set recordings targeted bats, they likely lack sounds often confused with bat calls, like those from small mammals (Coffey et al., 2019) or insects (Hall & Robinson, 2021). Further evaluation with real-world passive monitoring data and including potentially confounding sounds is necessary to obtain a more robust assessment of detection performance in ecologically relevant settings.

My work contributes to the exploration of strategies for addressing data scarcity in bioacoustic research. This work thus contributes to the broader field of learning from limited data, often called few-shot learning (Nolasco, Singh et al., 2023; Nolasco, Ghani et al., 2023; Song et al., 2023). While much of this research emphasises model improvements, I offer a complementary data-centric approach (Zha et al., 2023), focusing on enhancing the quality of the training data through detailed annotation. My results demonstrate that investing in detailed audio annotation can be a highly effective alternative to extensive data collection, particularly when field recording is challenging or cost-prohibitive. However, annotation itself can be a demanding process, highlighting the need for tools and methods that facilitate efficient and accurate annotation. The emergence of annotation tools specifically designed for machine learning development is a promising step towards bridging this gap (Chapter 2, Marsland et al., 2019). Additionally, active learning strategies can help optimise the annotation process by identifying the most informative data points for annotation, thereby reducing the overall annotation effort (Martinsson et al., 2024; McEwen et al., 2024; Wang et al., 2022). The growing availability of data platforms for sharing bioacoustic recordings presents another opportunity (Vellinga & Planque, 2015; Görföl et al., 2022; Matheson, 2014). By promoting the sharing of annotations

alongside recordings, these platforms can facilitate collaborative research and accelerate the development of robust machine learning models. However, standardised annotation formats are crucial to ensure transparency and interoperability. By developing methods to streamline the annotation process and promoting standardised data sharing, the potential within existing and future acoustic collections can be unlocked, ultimately accelerating bioacoustic research and conservation efforts.

Chapter 4

Enhancing Deep Learning for Bat Call Identification through Acoustically-Informed Architectures

4.1 Abstract

Acoustic monitoring is an effective and scalable way to assess the health of important bioindicators like bats. Deep Learning (DL) is increasingly used to automate bat echolocation call detection and classification, but developing these solutions for novel geographic regions is hindered by limited data and the lack of accessible tools. While current DL methods adapt techniques from computer vision, the fundamental differences between audio and visual data raise questions about their optimality, especially when data is scarce. Here, I develop BatDetect2, a novel, open-source DL pipeline for jointly detecting and classifying bat species from acoustic data. BatDetect2 adapts a Convolutional Neural Network (CNN), commonly used in image analysis, to detect bat echolocation calls within input spectrograms, with two key modifications for audio data. First, a self-attention layer is incorporated to capture long-range temporal dependencies within the echolocation call sequences. Second, the standard convolutional operation is modified to include the spectrogram's frequency coordinates, allowing the model to directly incorporate frequency position

into its calculations. The impact on performance of these modifications is evaluated on a UK dataset of 17 bat species, and the full BatDetect2 pipeline is further validated on five diverse datasets from the UK, Mexico, Australia, and Brazil. I found that adding the temporal modification increased the mean average precision (mAP) from 0.72 to 0.81, while the frequency modification had no notable impact. All tested DL models significantly outperformed a traditional call parameter extraction method, which achieved an mAP of 0.59. Overall, BatDetect2 showed strong performance across all datasets. This study demonstrates that the same pipeline can be applied, without modification, to acoustic data from diverse regions with varying species compositions. To the best of our knowledge, BatDetect2 is the first pipeline featuring a 2D convolutional architecture with a single, strategically placed temporal self-attention layer, designed to detect and classify all bat calls present in input spectrograms. The trained UK model and the full training pipeline are available through the open-source Python package, `batdetect2`. The model training and evaluation tools proposed will provide practitioners with an accessible means to develop models using their own data.

4.2 Introduction

Bats are vital bioindicators for assessing the impacts of climate change and habitat loss (Jones et al., 2009), yet significant knowledge gaps exist regarding the status of their populations (Frick et al., 2019). Acoustic monitoring, leveraging the use of echolocation by bats for navigation (Jones & Siemers, 2011; Prat et al., 2016), offers a scalable, non-invasive and cost-effective solution for studying their activity (Gibb et al., 2018). Considerable research effort has been dedicated to automating the detection and classification of bat echolocation calls in audio recordings (Zamora-Gutierrez et al., 2021), with methods evolving from the use hand-crafted acoustic features (Obrist & Boesch, 2018; Parsons & Jones, 2000) to recent Deep Learning (DL) approaches (Mac Aodha et al., 2018; Vogelbacher et al., 2023; Khalighifar et al., 2022). However, the practical scope of existing tools remains limited, restricted to specific species or regions, and frequently inhibited by proprietary restrictions that im-

pede transparency and accessibility. Furthermore, development is often constrained by the need for extensive datasets and specialised technical expertise (Stowell, 2022). Therefore, creating and understanding efficient methodologies for developing accurate bat detection and classification tools adequate for smaller datasets is crucial to facilitate broader bat monitoring and research (Russo et al., 2021).

Achieving accurate detection and classification of bat echolocation calls is challenging because bat calls are complex and varied. This variability stems from species-specific, regional, and habitat-dependent call characteristics (Walters et al., 2013; Montauban et al., 2021; Russo et al., 2018), which is further complicated by background noise and overlapping vocalisations from other species (e.g. small mammals and insects) (Stowell, 2022). The use of hand-crafted acoustic features, commonly referred to as call parameters in the bat literature, with traditional machine learning methods like Discriminant Function Analysis (Parsons & Jones, 2000) or Random Forest (RF) (Zamora-Gutierrez et al., 2021; Bas et al., 2017; Roemer et al., 2021) often struggle to adapt to this variability and to discriminate between similar-sounding species (Russo et al., 2018). In contrast, DL models can leverage more detailed inputs like spectrograms, potentially capturing overlooked but informative acoustic features. However, the numerous parameters that allow DL models to learn complex patterns also heighten the risk of overfitting, particularly with limited training data (Pichler & Hartig, 2023). Despite this potential for overfitting, DL models have achieved considerable success even with modest training datasets in broader ecological monitoring (Christin et al., 2019) and bioacoustics (Stowell, 2022). Nevertheless, the limited availability of data for most species and regions calls for careful consideration of how to balance model complexity with generalisation performance, and which DL architectures are best suited to address the unique characteristics of bat echolocation calls.

To date, all DL architectures applied to bat call detection and classification are adapted from the field of computer vision (Mac Aodha et al., 2018; Chen et al., 2020; Kobayashi et al., 2021; Zualkernan et al., 2020; Paumen et al., 2021; Dierckx et al., 2022; Schwab et al., 2022; Tabak et al., 2022; Yılmaz et al., 2022; Alipek et al., 2023;

Brinkløv et al., 2023; Fundel et al., 2023; Vogelbacher et al., 2023). Specifically, 2D Convolutional Neural Networks (CNNs) have been widely employed to analyse spectrograms or Mel-frequency cepstral coefficients (MFCCs) derived from audio recordings. The ability of CNNs to exploit the inherent properties of images, namely locality and translation invariance, through the convolutional operation allows for a more efficient architecture with significantly fewer parameters than fully-connected networks, leading to improved performance in image classification tasks (LeCun et al., 2015; Menghani, 2023). In the context of images, locality implies that information needed for object identification is spatially concentrated, while translation invariance means that an object's identity remains consistent regardless of its position within the image. However, these core assumptions in computer vision do not necessarily translate well to the analysis of audio data.

Bat echolocation calls exhibit unique spectro-temporal characteristics that challenge the direct application of standard CNN architectures. For instance, the distinct frequency ranges of calls emitted by different bat species, which reflect adaptations to their foraging environments and prey types (Denzinger & Schnitzler, 2013; Walters et al., 2013), imply that translation invariance may not hold in the frequency dimension for bat calls. Furthermore, as bats typically emit sequences of echolocation calls, where the inter-pulse interval and the overall temporal structure are often crucial for accurate species identification, the assumption of locality might not be entirely appropriate in the temporal dimension. While analysing short audio clips, typically shorter than 50 milliseconds, reduces the impact of this issue (Mac Aodha et al., 2018; Chen et al., 2020; Kobayashi et al., 2021; Khalighifar et al., 2022), it leaves out the potentially discriminative information encoded in the longer temporal structure of call sequences. Using CNNs to analyse longer audio clips, as done by Paumen et al. (2021), Zualkernan et al. (2020) and Tabak et al. (2022), typically requires increasing model depth and size to capture longer temporal relationships (Simonyan & Zisserman, 2015; He et al., 2016), thereby demanding substantially more training data for robust results. This naturally raises the question of how to design a model that can effectively capture both the spectral and temporal structure of bat

echolocation calls while remaining compact and efficient.

Despite the recent progress in DL-based solutions for bat detection, a gap persists between the latest research advancements and the open-source tools available to practitioners. Of the DL models developed for automated bat detection, only Mac Aodha et al. (2018), Alipek et al. (2023) and Fundel et al. (2023) offer open-source implementations, but require substantial programming expertise to use. Furthermore, training and using custom DL models presents several additional challenges. First, even with the availability of open-source tools for developing bioacoustic models (Lapp et al., 2023), training models typically requires proficiency in programming and machine learning, presenting a barrier for many practitioners. Secondly, training DL models requires substantial amounts of labelled data, which are often scarce for many bat species and regions. Although transfer learning, where a model pre-trained on a large dataset is fine-tuned on a smaller, more specific dataset, can help to mitigate data scarcity (Ghani et al., 2023; Dufourq et al., 2021), it does not eliminate the need for expertise in model training. Finally, the "black box" nature of many DL models makes it difficult to interpret their decision-making processes. This lack of transparency is particularly problematic when a model generates a single prediction from a long audio clip containing multiple calls from different species (Dierckx et al., 2022). Moreover, processing audio at the clip level, rather than analysing individual calls, prevents the use of fine-grained call information that might improve performance (Chapter 3). Given these challenges, there is a clear need for user-friendly tools that empower practitioners to develop and deploy robust DL models for bat monitoring, without requiring extensive programming or machine learning expertise.

Here, I develop BatDetect2, a novel model for bat echolocation call detection and species classification from acoustic data. BatDetect2 incorporates two key modifications to the standard CNN architecture, specifically designed to enhance the model's ability to better capture spectral and temporal characteristics. Furthermore, the model also provides interpretable predictions that illustrate where in the input spectrogram, in terms of frequency and time, the model has detected a call. Using

a UK dataset of 17 bat species, I evaluate the impact of these modifications on the model's overall performance, as well as on its performance for each individual species. I train and evaluate the model using five challenging datasets from four different geographical regions (UK, Mexico, Australia, Brazil), and compare its performance to existing call parameter-based methods. Using the model trained on UK data I evaluate its potential for transfer learning to other regions. Finally, to facilitate adoption and further development by practitioners, the complete pipeline, including code and trained models, is made publicly available as an open-source Python package: `batdetect2`. This package enables users to train new models from scratch, fine-tune pre-trained models, and deploy them for automated analysis of their own datasets.

4.3 Materials and Methods

4.3.1 Acoustic event detection and classification

Distinct acoustic vocalisation events (e.g. a bat echolocation call or a bird song) created by a species of interest can be characterised by the start time of the event, the duration of the event, and the minimum and maximum frequency bands that the event spans. The goal of this work is to develop a model, denoted by $g(\cdot)$, that takes an ultrasonic audio recording as input, represented as a spectrogram $\mathbf{x}$, and outputs a set of predictions related to the events of interest in the input audio file, $\mathcal{O} = g(\mathbf{x})$. In this specific context, the events of interest are bat echolocation calls. Each prediction from the model, $\mathbf{o} \in \mathcal{O}$, represents a distinct event and provides information about its temporal and spectral characteristics, as well as its predicted species. Specifically, each predicted event, $\mathbf{o} = [t_{\text{start}}, t_{\text{end}}, f_{\text{min}}, f_{\text{max}}, \mathbf{p}_{\text{species}}]$, represents the start time, end time, minimum frequency, maximum frequency of the event, along with probability vector indicating which species the model thinks is present. Here, $\mathbf{p}_{\text{species}}$ is a $C+1$ dimensional vector that sums to one, and represents the probability of the species the model thinks emitted the call, for each one of C different species of interest plus one additional background class (i.e. 'Not bat'). Note, that this representation is distinct from conventional acoustic classification models that only attempt to determine the

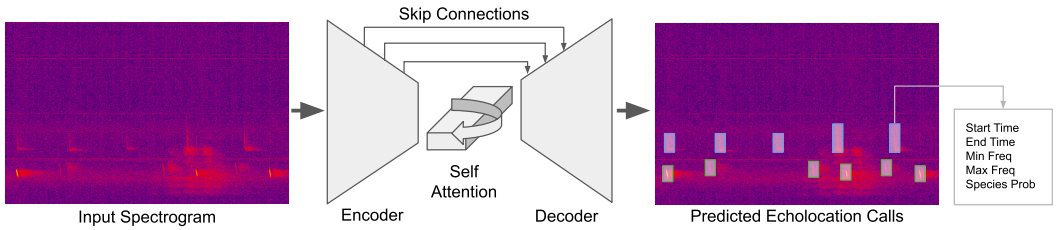


Figure 4.1: Overview of BatDetect2 architecture. The model consists of a convolutional neural network-based encoder and decoder with skip connections that share extracted features from the encoder to decoder. The encoder and decoder consists of modified convolutional layers that incorporate coordinate information. It utilises a self-attention layer in the middle of the model so that it can reason over a longer temporal scale. In contrast to most existing deep learning-based bat call classifiers, BatDetect2 directly predicts the time in file of each event of interest, along with the duration of the event, the frequency range, and the species.

species present in a short duration input spectrogram (Stowell, 2022), i.e. $y = g(\mathbf{x})$, where $y \in \{1, ..., C+1\}$ is an integer denoting the predicted species label.

4.3.2 Model architecture

I implement the joint classification and detection model $g(\cdot)$ as a deep neural network, which I refer to as BatDetect2. Inspired by computationally efficient one-stage object detection methods from computer vision, e.g. Zhou et al. (2019), this model directly predicts the location and size of each event (i.e., echolocation call) in the input. BatDetect2 model makes use of a U-Net-style architecture (Ronneberger et al., 2015), with an encoder that extracts features from the input spectrogram, followed by a decoder that generates the predicted size and location of each echolocation call along with the corresponding species' probabilities (Figure 4.1). The model also incorporates skip connections, which facilitate the propagation of higher-resolution feature information (in terms of frequency and time) from the encoder to the decoder (see Appendix C.1 Table C.1 for full details). Crucially, while BatDetect2 is based on the U-Net architecture, which typically employs 2D convolutional layers, it incorporates two key modifications to better capture the distinctive spectral and temporal characteristics of audio signals described below.

In order to allow the model to capture the temporal structure of the echolocation calls, I incorporate a self-attention layer into the middle of the network (Figure 4.1). Transformer-based self-attention architectures (Vaswani, 2017) are among the current

best performing models in natural language processing owning to their ability to capture long-range dependencies that occur in the input data. The introduction of this layer allows the model to 'attend' to information from different points in time in the input audio file in order to increase or decrease its estimated likelihood that a given species is present at the current time step. BatDetect2 processes a spectrogram input with a temporal resolution of approximately 1ms per time bin (1024 Hz). The encoder component transforms this input into a sequence of feature vectors at a reduced temporal resolution of approximately 8ms per bin (128 Hz). For each time step in this coarser representation, the self-attention layer uses transformations of the encoded feature vector to determine its relationship to all other time steps. These relationships generate attention weights, which are then used to compute a weighted average of transformed feature representations. This weighted average provides a context-aware representation for that time step. Finally, the decoder component processes these context-aware representations to generate an output with the same temporal resolution as the original spectrogram, including the model's predictions for the locations of bat calls. Note that, unlike vision transformers such as ViT (Dosovitskiy et al., 2021) used by Fundel et al. (2023), which employ multiple self-attention layers across both time and frequency, our model uses a single self-attention layer operating solely on the temporal dimension, resulting in significantly reduced computational burden.

To enhance the ability of BatDetect2 to process frequency information and mitigate the undesirable translation invariance along the frequency axis, I utilise two specialised building blocks: `CoordConvDown` and `CoordConvUp`. At a high level, the `CoordConvDown` layer takes a tensor as input and returns a spatially downsized version of it as an output. Unlike standard convolutional layers, which exhibit translation invariance, `CoordConvDown` appends non-learnable, normalised coordinates along the frequency (vertical) axis of the input tensor. This modification is crucial because the absolute frequency of an echolocation call provides valuable discriminative information for bat species identification. In contrast to the original `CoordConv` approach (Liu et al., 2018), coordinate information is not added along the temporal

axis. This design choice preserves the model's desired time-translation invariance, allowing it to recognise calls regardless of their precise position within the recording. The `CoordConvUp` layer performs the inverse operation of `CoordConvDown`, upsampling the feature maps while retaining the encoded frequency information.

For each input spectrogram, the model initially generates an intermediate "feature" map, an array with the same height and width as the input. Each pixel in the feature map encodes a 32-dimensional feature vector, representing learned acoustic characteristics at the corresponding time-frequency location. This feature map is then used to produce two primary outputs: a "class" map ($\hat{Y}$) and a "size" map ($\hat{S}$). Both of these outputs have the same height and width as the input spectrogram.

The class map $\hat{Y}$ indicates where the model predicts echolocation calls are located within the spectrogram and what species they belong to. Each pixel in $\hat{Y}$ contains a vector with a length equal to the number of bat species in the dataset plus one. This additional element represents a "background" class, indicating the absence of a call. Each value in the vector represents the model's confidence that a call of a particular species is present at that pixel's corresponding time and frequency. Ideally, only the pixel corresponding to the bottom-left corner of a call's bounding box should have a high confidence value for the correct species, and all other pixels should indicate "background." The size map, $\hat{S}$, provides information about the estimated size of the detected echolocation calls. Each pixel in $\hat{S}$ contains two values: an estimated height and width of the bounding box around any echolocation call detected at that pixel's location. If the model does not detect a call at a particular location, the corresponding values in $\hat{S}$ should ideally be zero.

As a final step, this output is pass through a non-maximal suppression layer, implemented via max pooling, in order to extract the local peak detections (Zhou et al., 2019). This step prevents the model from predicting multiple calls very close to each other (i.e. within a few milliseconds).

4.3.3 Audio preprocessing

To prepare the raw audio for model processing, the input audio is transformed into spectrograms as follows. Firstly, the input audio is resampled to 256 kHz using the polyphase method from `librosa` (McFee et al., 2015). I selected `librosa`'s polyphase resampling method for its comparatively lower computational cost, making it well-suited for processing large audio datasets. I then compute the magnitude spectrogram using a Short Time Fourier Transform (STFT) with a window size of 512 samples and a window overlap of 75%. As the bat echolocation calls recorded for this study are found only within a specific frequency range, I retain only the bands between 10 kHz and 120 kHz. For robustness to volume variations, the spectrogram is normalised using Per-Channel Energy Normalisation (PCEN) (Wang et al., 2017), which Lostanlen et al., 2019 showed to be more effective than traditional logarithmic-based normalisation. Following Aide et al. (2013) and Mac Aodha et al. (2018), I also subtract the mean value from each frequency band to mitigate the impact of any constant background noise. Finally, I use bilinear interpolation to resize the temporal dimension down by a factor of two and resample the frequency bands into 128 bins. Consequently, a one-second input audio file results in a spectrogram of size 128×1024.

4.3.4 Model training

The model is trained using a supervised learning approach, where it is provided with input spectrograms and corresponding target outputs. These target outputs are derived from the bounding box annotations of bat echolocation calls. The target for the "class" map $\hat{Y}$ is constructed by creating a series of "heatmaps," one for each species. Each species-specific heatmap is initially set to zero everywhere except in the vicinity of the bottom-left corner of each call annotated for that species. At these locations, a Gaussian kernel with a standard deviation of 2.0 is applied, creating a localized peak with smoothly decaying values, a method consistent with prior work in object detection (Zhou et al., 2019). Unlike Zhou et al. (2019), who parameterises bounding boxes using their centre point, I instead use the point corresponding to the start time and minimum frequency of each echolocation call. I chose the minimum

frequency point because it exhibited less inter-species variability compared to the center frequency across the training data, with the notable exception of the two UK *Rhinolophus* species. Once the individual species-specific heatmaps are generated, they are stacked together to form the final target map for $\hat{Y}$. To create the target for the "size" map $\hat{S}$, the height and width of the bounding box of each annotated call are calculated in pixel units. These values are then assigned to the pixel location corresponding to the bottom-left corner of the bounding box. All other pixel locations, which correspond to areas without annotated calls, are assigned values of zero for both height and width.

The model is trained end-to-end using a three component loss function which includes a detection loss, a classification loss, and an event size loss. The detection loss is computed by comparing the predicted class map ($\hat{Y}$) to the ground truth class map (Y), considering only the complement of the background class (i.e., the sum of species-specific heatmaps). In contrast, the classification loss compares all individual species-specific heatmaps. Both losses are implemented using a focal loss, as described by Lin et al. (2017). This loss function is particularly well-suited to this scenario as most spectrogram pixels do not contain calls, resulting in a substantial class imbalance between "background" and "call"' pixels. The event size loss is calculated using an L1 loss that penalises the absolute difference between predicted and actual dimensions, thus encouraging accurate size estimations. This loss is computed only for pixels corresponding to true calls (see Appendix C.2 for a detailed description of the training losses).

To increase the variation in the input audio, I perform a series of augmentations at training time. These augmentations include: random linear combination of two input audio files (Zhang et al., 2018), simulated echo, random volume scaling, temporal stretching, and time and frequency masking (Park et al., 2019). The probability that any one augmentation is applied is 0.2, and multiple augmentations can be applied simultaneously.

The model and training code are implemented in `PyTorch` (Paszke et al., 2019). I

train the model end-to-end using the Adam optimizer (Kingma & Ba, 2017), starting with an initial learning rate of 0.001, a cosine annealing learning rate schedule, and a batch size of 8. Training is done for 200 epochs.

4.3.5 Audio datasets

I train and evaluate the model on five different full spectrum ultrasonic acoustic datasets. In preparation for training, these datasets require annotations in the form of bounding boxes that encompass each individual echolocation call within an audio file. To generate these annotations, an early version of `whombat`, the audio annotation tool described in Chapter 2, was employed. Unless otherwise specified, the annotated audio files had information at the file-level related to which species were present in the recording. The annotations were created by a team of bat experts and myself.

Annotators were instructed to draw boxes around each individual echolocation call, irrespective of how faint the call was. They then assigned the recording-level species class label to an annotation unless it differed from a prototypical echolocation call for that species. Harmonics were not annotated as part of the main call. In cases where it was not possible to assign the correct class label, or when multiple species were present in a file, annotators marked unknown calls as being from a generic 'Bat' class. Additional details for each dataset, including per-species counts, are available in the Appendix C.3.

4.3.5.1 UK datasets

The primary dataset used in this study comprises recordings of 17 bat species known to breed in the UK. This dataset was collated from six distinct sources, including the Bat Conservation Trust and individual contributors, ensuring a wide variety of recording devices and acoustic environments. This diversity is important as it maximises the variation in the training set, with the ultimate aim of having better generalisation performance at test time. The vast majority of the recordings were made in the UK, but there were also some additional files included from the species of interest that were recorded elsewhere (e.g. Europe). In total, the dataset contains 2,809 distinct audio files with a mean duration of 1.04 seconds, encompassing 34,635

annotated echolocation calls.

To increase the robustness to background noise, I supplement this data with 4,225 additional 0.384-second duration files adapted from Mac Aodha et al. (2018) and collected in the iBats Program (Jones et al., 2013). This adds an additional 6,842 annotated bat calls that do not have a confirmed species label. Finally, I also add 345, one second duration, empty files (i.e. no bats present) from London, UK, collected using the recording devices described in Gallacher et al. (2021). These "empty" files enable the model to learn to better distinguish between bat calls and background noise.

I split the UK data into two different train and test sets, UK_{same} and UK_{diff}. For UK_{same} I randomly assign files to the test set, ensuring a maximum of four files per species, per data source. The remaining files are kept for the training set. This results in 7,010 train files and 369 test files, containing 36,955 and 4,522 calls respectively. UK_{diff} is a more challenging split. Here I hold-out the largest single data source for testing. This leaves 5,911 training and 1,468 test files, containing 24,315 and 17,162 echolocation calls. This second split represents a more challenging test-case where the data is guaranteed to be very different from the training set. This also results in a reduction in the overall amount of training data, both in terms of sheer quantity but also diversity. Both the UK_{same} and UK_{diff} training sets include the 4,570 files without species labels. The average echolocation calls for each species in this dataset are visualised in Appendix C.1.

4.3.5.2 Yucatan data

The second dataset consists of 1,193 one second audio clips extracted from 285 passive acoustic recordings from the Yucatan peninsula in Mexico. The data was collected as part of a study by MacSwiney G. et al. (2008). It is smaller in size than the UK dataset, but is representative of the type of data that would be feasible to collect and annotate as part of a smaller-scale monitoring project. The annotations from the original study were used and then expanded to ensure that all audible echolocation events were annotated. The final annotated dataset contains 10,020

echolocation calls from 17 different species. I divided the data into 911 training and 282 test clips, making sure to separate at the original recording-level, and not the clip-level, to ensure that clips from the same recording were not in both sets. The average echolocation calls for each species in this dataset are visualised in Appendix C.2.

4.3.5.3 Australia data

This next dataset consists of a set of 14 bat species which can be found in the major cotton growing region on the north west plains of New South Wales and adjacent areas in central southern Queensland. Bat calls were recorded in the field from individuals released after capture, following positive species identification. This dataset features species with similar call characteristics which makes it particularly challenging. The data was randomly split at the file level, with 80% of the recordings for a species staying the train set, and the rest in the test. This resulted in 4,569 and 1,327 individual calls in the train and test sets respectively. The average echolocation calls for each species in this dataset are visualised in Appendix C.3.

4.3.5.4 Brazil data

The final dataset presents a distinct challenge as it lacks confirmed species labels. It contains 320 recordings of ten second duration each collected between January and March 2019 in south-eastern Brazil using AudioMoth recorders (Hill et al., 2019). As the identity of recorded bat species could not be independently verified, calls could not be assigned species labels during annotation. Instead, I created three 'sonotypes' based on the dominant frequency component of each call and labeled individual calls accordingly. Like the other datasets, the annotation was performed manually, where the protocol again stipulated that all echolocation call instances in each recording should be annotated. I split the data into 256 train files and 64 test files, which resulted in 7,989 and 2,010 calls respectively. The average echolocation calls for each sonotype in this dataset are visualised in Appendix C.4.

4.3.6 Evaluation metrics

In order to evaluate model performance, I use four different evaluation metrics. The first, detection average precision ('AP Det'), evaluates the ability of the model to correctly identify all valid echolocation calls in the test data. This metric calculates the precision and recall resulting from varying a threshold on the model output predictions for the 'Bat' versus 'Not bat' task. I then average over these different thresholds to quantify the area under the precision-recall curve, using the interpolation method used in Everingham et al. (2009). A prediction is counted as a true positive if its estimated start time overlaps with a ground truth echolocation call by at most ten milliseconds. This is the same evaluation criteria used in Mac Aodha et al. (2018).

'AP Det' does not evaluate the ability of the model to accurately assign the correct species label to a prediction. To address this, I also report the mean average precision across the classes ('mAP Class'). This involves taking the per-class average precision and then averaging this value over each class. This also has the added effect of weighting each class equally, irrespective of the number of calls for each class in the test set. Here, I exclude calls for which there are no ground truth species labels available.

'mAP Class' suffers from one major limitation. As the classes are evaluated independently, it does not highlight cases where the underlying model may be poorly calibrated and thus require different output thresholds for each class. Calibration issues like this can result from class-level data imbalances in the training data. To overcome this limitation, I also report a third precision based metric which I refer to as 'Top Class'. Here I simply take the top predicted class label, along with its corresponding probability, for each detected call and then evaluate the average precision as above. Unlike 'mAP Class', this metric can be biased if there is a large imbalance in the classes in the test set.

The final metric, 'File Acc', evaluates the file-level classification accuracy. For this metric only, test files manually annotated as containing multiple species are

excluded. To obtain a single file-level class label from the multiple individual call predictions within a given file, each detection is thresholded, and those below the threshold are removed. Multiple thresholds are evaluated, and the single threshold that yields the best overall performance across all files for a given model is selected. I then sum the per-class probabilities of the remaining detections and choose the class with the highest sum as the file-level prediction. Finally, I report the file-level accuracy corresponding to the single best threshold across all files. The best possible score for each of these four metrics is 1.0, and the worst is 0.0.

4.3.7 Experiments

4.3.7.1 Architecture modifications experiments

To assess the impact of the proposed architectural modifications on model performance, I conducted an ablation study. In addition to the full BatDetect2 model, I trained and evaluated two model variants: (1) NoSelfAttn, a variant without the self-attention layer, and (2) NoCoordConv, a variant where the `CoordConvUp` and `CoordConvDown` layers are replaced with standard convolutional layers. All three models are trained and evaluated on the challenging UK_{diff} dataset split using identical training and evaluation protocols.

To analyse performance differences at the species level, I compute per-class average precision (AP) each species in the UK_{diff} dataset, in addition to the standard global evaluation metrics. This allows for a more granular analysis of how each architectural modification affects the detection and classification accuracy for individual bat species.

4.3.7.2 Comparison with call parameter baseline

To evaluate the model's effectiveness across diverse settings, I compare it to a baseline traditional bat call parameter extraction method. Both the BatDetect2 model and the baseline are trained on all five datasets (UK_{same}, UK_{diff}, Yucatan, Brazil and Australia) using identical training and testing data and are evaluated using the same metrics. The BatDetect2 model is trained three times on each dataset to account for stochastic fluctuations in the training process, and the final results are averaged.

To train the baseline, I use the Tadarida-D model from Bas et al. (2017), which consists of two main components: (i) a bat echolocation call detector and (ii) a echolocation call feature extractor. For a given training dataset, I run Tadarida-D on each recording producing detected calls with extracted call features. These features are a set of 268 numerical values that encode information about the shape and frequency content of each detected call (see Bas et al., 2017 for further details). Then for each detected event, I compute the overlap between the event (using the reported time in file, duration, and frequency range from Tadarida-D) and the ground truth annotations. The detection with the highest overlap to a given ground truth annotation is assigned the corresponding species label. If a detected event does not match to a ground truth annotation it is assigned to the 'Not bat' class. Each ground truth annotation can only be assigned to one predicted detection. Finally I train a Random Forest (Breiman, 2001) classifier on the extracted calls using the implementation from `scikit-learn` (Pedregosa et al., 2011) with default parameters. I employed a Random Forest classifier to maintain consistency with the methodology of Bas et al. (2017), facilitating comparison of results.

This baseline allows for a controlled comparison with a traditional call parameter-based method by using the same audio data and ground truth annotations for both training and evaluation. However, it is important to emphasise that while I am using Tadarida-D, the baseline is *not* directly equivalent to the full Tadarida method as I do *not* make use of their pre-trained models, labeling interface, classification code, or post-processing steps.

4.3.7.3 Transfer learning evaluation

To evaluate the transfer learning potential of BatDetect2, I assessed its performance on three datasets (Yucatan, Brazil, and Australia) using three different model variants: $\text{BatDetect2}_{\text{zero}}$, $\text{BatDetect2}_{\text{tuned}}$ and $\text{BatDetect2}_{\text{full}}$. $\text{BatDetect2}_{\text{zero}}$ refers to the base model pre-trained on the UK_{same} dataset and applied directly to the target datasets without any further training or modification. $\text{BatDetect2}_{\text{tuned}}$ represents a fine-tuned model where a Logistic Regression classifier is trained on features extracted by the pre-trained BatDetect2. Finally, $\text{BatDetect2}_{\text{full}}$ is the BatDetect2 model trained

from scratch on each individual target dataset and is included for comparison with the transfer learning approaches. The UK_{same} dataset was selected as the source for pre-training due to its larger size and greater diversity compared to the other datasets, providing a robust foundation for a generalisable model. This evaluation focuses on two key aspects: (1) the off-the-shelf bat call detection performance of $BatDetect2_{zero}$ in novel regions with unseen species and (2) the effectiveness of the learned feature embeddings for species classification in these new contexts, using a model trained with these features, $BatDetect_{tuned}$.

The evaluation procedure closely follows the methodology used in the RF + Tadarida-D baseline comparison. The pre-trained BatDetect2 model was applied to each of the three target datasets (Yucatan, Brazil, and Australia). For each dataset, I registered all detected calls and extracted the corresponding 32-dimensional feature embeddings from the "feature" map. The "feature" map represents the output of the decoder before the final classification and size prediction layers, thus capturing the model's learned representation of bat calls. Similar to the RF + Tadarida-D baseline, detected calls were matched to ground truth annotations using a simple overlap criterion, specifically, a 10 ms overlap between the predicted start time and the ground truth start time. To assess the generality of the bat detector, I computed the detection average precision (AP Det) using the generated detections and their associated confidence scores on the corresponding test set.

To evaluate the transferability of the learned feature embeddings, I trained a logistic regression classifier, denoted as $BatDetect2_{tuned}$ on the extracted features. This approach is conceptually similar to that used by Ghani et al. (2023), but with a key distinction. Ghani et al. (2023) focused on embeddings representing the entire input spectrograms, whereas BatDetect2 produces potentially multiple detections per input spectrogram, each with its own corresponding localised feature embedding. Therefore, the logistic regression is trained to classify individual detected calls based on their localised features, rather than classifying entire audio clips. The classifier was implemented using the 'LogisticRegression' class from the 'scikit-learn' library (Pedregosa et al., 2011) with default parameters, The model was trained using

features extracted from the training set recordings of each target dataset and then evaluated on the corresponding test set recordings. The classification performance was quantified using the mean average precision across all classes (mAP Class).

4.4 Results

4.4.1 Impact of architectural modifications

I found that including the self-attention mechanism in the model resulted in a substantial improvement in performance (Table 4.1). Specifically, while call detection (measured by AP Det) remained unaffected, classification accuracy improved considerably. The mAP Class and Top Class metrics increased markedly from 0.725 to 0.810 and from 0.614 to 0.690, respectively, upon inclusion of the self-attention layer. On the contrary, the inclusion of coordinate convolutional layers yielded no discernible impact on performance.

Table 4.1: Performance of BatDetect2 variants on the UK_{diff} test set. Metrics are the average precision for detection (AP Det), the mean average precision for classification (mAP Class), the top class accuracy (Top Class), and the file accuracy (File Acc). The 'NoSelfAttn' variant is identical to the full BatDetect2 model but omits the self-attention layer. The 'NoCoordConv' variant is identical to the full model but does not incorporate frequency coordinate information into the convolutional layers. Each model was trained three times on the UK_{diff} training split; the mean performance on the corresponding test split is reported.

Model	AP Det	mAP Class	Top Class	File Acc
Full model	0.964	0.810	0.690	0.780
NoSelfAttn	0.962	0.725	0.614	0.790
NoCoordConv	0.960	0.811	0.681	0.774

Analysing the impact of self-attention on a per-species level reveals a consistent positive effect of the inclusion of self-attention across all species (Figure 4.2A). The improvement in average precision is particularly noticeable for species that exhibited lower performance without the self-attention. For instance, the model's ability to classify the *Myotis* genus, which encompasses several challenging and previously poorly-performing species, was significantly improved. In contrast, removing the *CoordConv* layers resulted in a negligible impact on overall performance across all species, although a slight improvement was observed for the two lowest-performing *Myotis* species (Figure 4.2B). Notably, the performance disparities between indi-

vidual species do not correlate with the amount of training data available for each (Figure 4.2C). This suggests that the self-attention mechanism's contribution lies in its ability to enhance discrimination between species with similar call characteristics, rather than improving the model's capacity to learn from larger datasets (see Appendix C.5 Figure C.6 for an illustrative example).

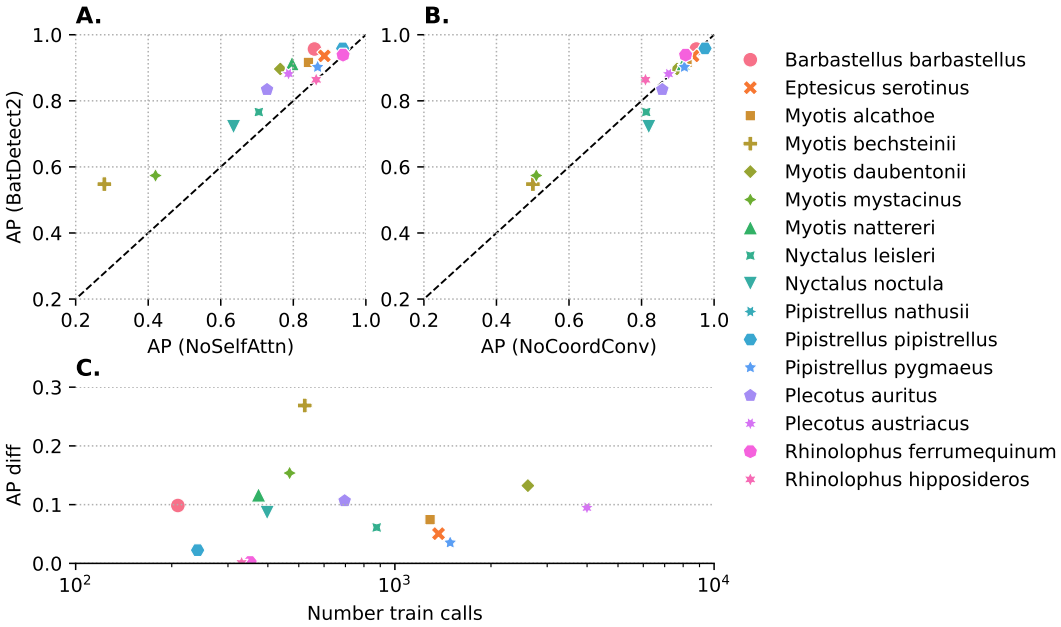


Figure 4.2: Impact of BatDetect2 modifications on per-species average precision. (A) Average precision (AP) for each species in the UK_{diff} dataset for the full BatDetect2 model and the NoSelfAttn variant (without self-attention). Points above the diagonal line indicate superior performance of the full model. (B) AP for each species in the UK_{diff} dataset for the full BatDetect2 model and the NoCoordConv variant (without frequency information added to the convolutional layers). (C) Difference in AP between the full BatDetect2 model and the NoSelfAttn variant (y-axis) versus the number of echolocation calls in the UK_{diff} training dataset (x-axis) for each species. All models were trained three times on the UK_{diff} training split, and the mean AP on the corresponding test split is reported.

4.4.2 Detection and classification performance

The full BatDetect2 shows a substantial improvement over the Random Forest (RF) baseline across all datasets and evaluation metrics (Table 4.2). The performance difference, measured by mean average precision (mAP Class), ranges from 0.05 to 0.37 across the datasets. While the RF baseline achieves reasonable performance on the comparatively less complex Brazil dataset, it exhibits significantly lower performance on the remaining datasets.

Table 4.2: Performance of BatDetect2 model compared to the Random Forest baseline with traditional bat echolocation call features. Both models are evaluated using the same five test datasets. For each of the metrics, higher numbers are better, and the results are averaged over three runs.

	BatDetect2				**Random Forest Baseline**			
Dataset	AP Det	mAP Class	Top Class	File Acc	AP Det	mAP Class	Top Class	File Acc
UK_{same}	0.971	0.884	0.843	0.866	0.890	0.706	0.638	0.800
UK_{diff}	0.964	0.810	0.690	0.780	0.903	0.587	0.47	0.687
Yucatan	0.923	0.803	0.818	0.861	0.649	0.430	0.467	0.682
Australia	0.973	0.700	0.640	0.795	0.928	0.603	0.507	0.719
Brazil	0.926	0.962	0.940	1.000	0.883	0.912	0.910	1.000

The detection performance of the full BatDetect2 model, as indicated by the AP Det metric, reveals consistent results above 0.92 in all cases (Table 4.2). This suggests that the model successfully detects the majority of bat calls present in the data. BatDetect2 appears to be robust to background noise, as even in the presence of repetitive high-frequency noise, or sudden broad band clicks, the model does not produce false positives (Figure 4.3).

On the contrary, the call-level classification performance, measured by mean average precision (mAP) and top class average precision (Top Class), showed considerable variability across datasets, ranging from 0.70 (mAP) and 0.64 (Top Class) for the Australia dataset to 0.96 (mAP) and 0.94 (Top Class) for the Brazil dataset. The performance discrepancy between the UK_{diff} and UK_{same} datasets likely reflects the more rigorous train-test split employed in the former, coupled with the associated reduction in training data size. This split results in a test set that is less similar to the training data, despite covering the same species, thus posing a greater challenge for generalisation. The comparatively lower performance on the Yucatan and Australia datasets can partially be explained by the challenging set of species contained within each, as well as the smaller number of distinct training files available.

Examining the performance of the full BatDetect2 model on the challenging UK_{diff} dataset split at the species level shows challenges in accurately classifying certain *Myotis* species (Figure 4.4A). Notably, the precision for *Myotis bechsteinii* and *Myotis mystacinus* remains below 0.7 across all threshold levels. While *Nyctalus*

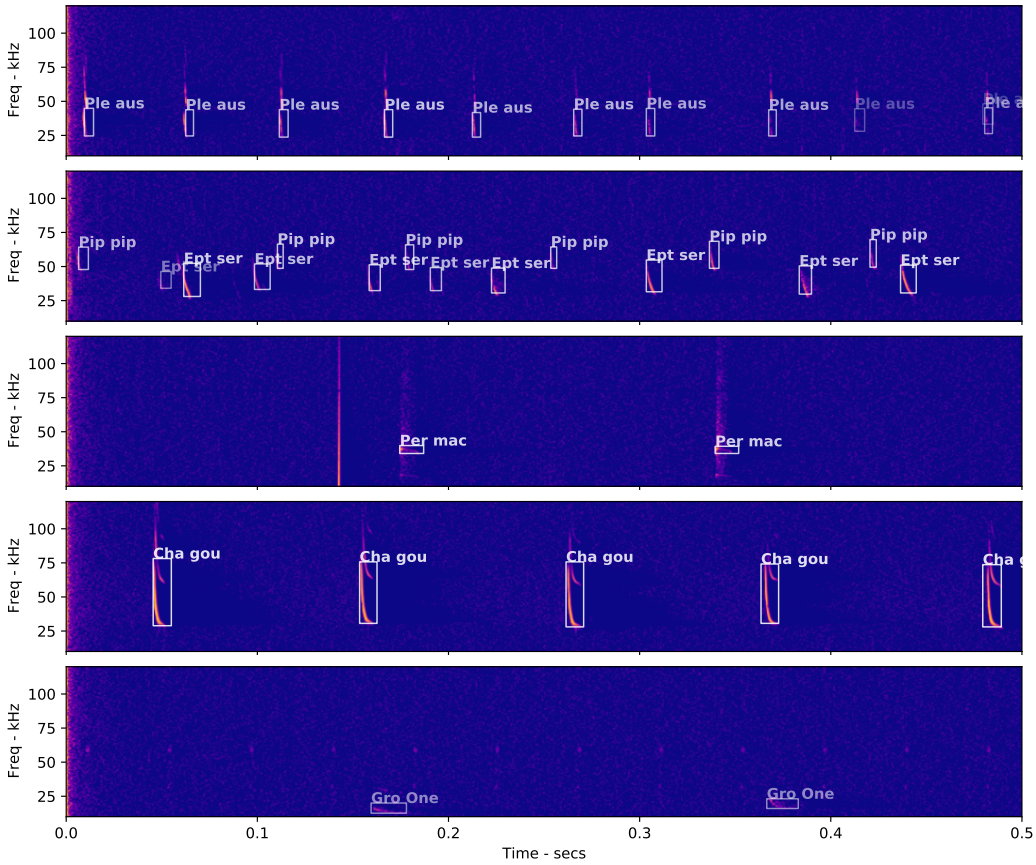


Figure 4.3: Predictions from the BatDetect2 model. Each row represents a different audio file selected from the test sets of the UK_{same}, UK_{diff}, Yucatan, Australia, and Brazil datasets, ordered from top to bottom. The intensity of an individual predicted bounding box indicates the model's confidence, with a brighter white value indicating more confident. The text above each box corresponds to the highest probability class label.

leisleri and *Nyctalus noctula* exhibit precision above 0.8 at a recall of 0.6, their performance declines at higher recall values. In contrast, all other species achieve a precision of at least 0.8 at a recall of 0.8. Analysis at the file level indicates considerable inter-species confusion within the *Myotis* genus (Figure 4.4B). Furthermore, when excluding the poorly-performing *Myotis* species, no correlation between the number of training examples and average precision is apparent (Figure 4.4C). This suggests that factors beyond training data size may contribute to the observed performance variations.

It takes BatDetect2 just under four minutes to process and save the results for 424, ten second duration, 384kHz AudioMoth recordings using a GPU, i.e. 70.6

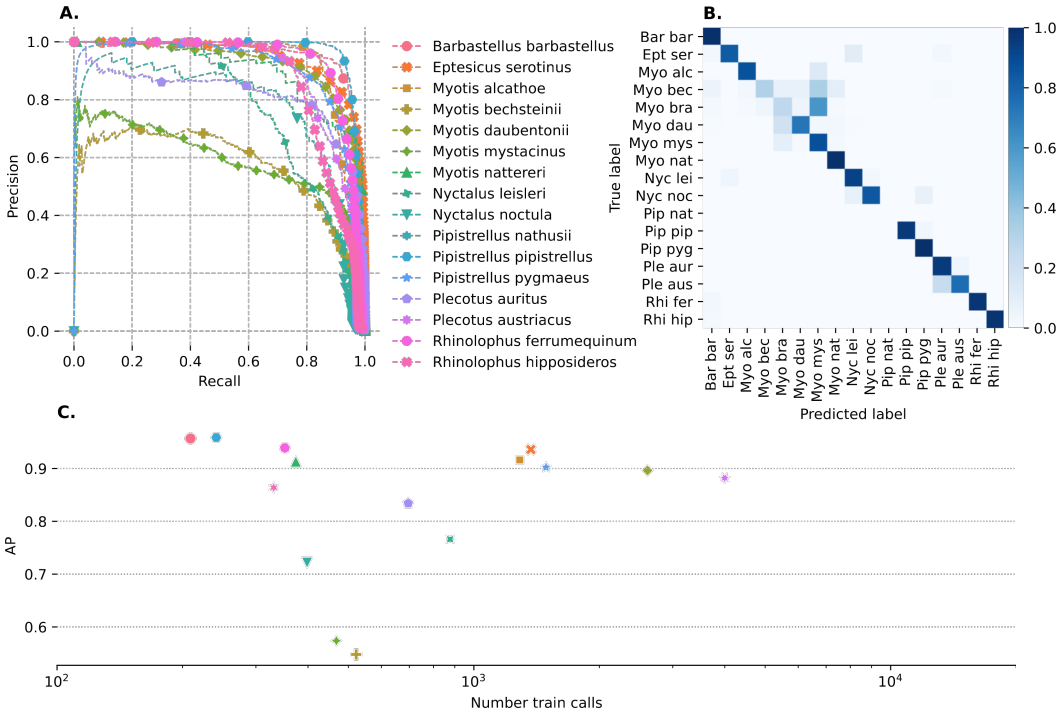


Figure 4.4: Per-species performance of BatDetect2 on the UK_{diff} dataset. (A) Precision-recall curves for each species in the UK_{diff} dataset. (B) File-level confusion matrix for the UK_{diff} dataset. The confusion matrix is computed by assigning each file to the species with the highest predicted probability and comparing this assignment to the ground truth species label. Rows are normalized to sum to 1. (C) Average precision (AP) for each species in the UK_{diff} dataset plotted against the number of echolocation calls in the training set.

minutes of ultrasonic data in total. Tadarida-D, which does not utilise a GPU, takes 2.5 minutes for detection and feature extraction for the same data. Note that this processing time does not include the evaluation of the RF. This benchmarking was performed on a workstation which contained an Intel i7-6850K CPU and an Nvidia TITAN Xp GPU.

4.4.3 Transfer learning performance

I found that $BatDetect2_{zero}$, the model trained solely on the UK_{same} dataset and applied directly to the target datasets, exhibited varied detection performance. While it achieved a high detection average precision (AP Det) of 0.921 on the Australia dataset, its performance was significantly lower on the Brazil dataset (AP Det = 0.650; see Table 4.3). In contrast, $BatDetect2_{tuned}$, which incorporates a Logistic Regression classifier trained on features extracted by the pre-trained BatDetect2, demonstrated

consistently improved detection performance across all datasets, with AP Det values exceeding 0.830. In general, both BatDetect2$_{zero}$ and BatDetect2$_{tuned}$ outperformed the Random Forest baseline in terms of AP Det, with the exception of the Brazil dataset where Tadarida-D achieved a higher AP Det of 0.883. BatDetect2$_{full}$, the model trained from scratch on each target dataset, consistently achieved the highest AP Det scores across all datasets.

Table 4.3: Evaluation of BatDetect2 transfer learning performance to three target regions. Models are evaluated using Average Precision (AP Det) for detection and mean Average Precision (mAP Class) for classification. BatDetect2$_{zero}$, trained solely on UK$_{same}$, is applied without further training to the target datasets; classification is not evaluated due to species differences. BatDetect2$_{tuned}$ uses the pre-trained BatDetect2 for detection and feature extraction, with a Logistic Regression classifier trained on these features for species prediction. BatDetect2$_{full}$ is trained from scratch on each target dataset. The performance metrics presented here for this model and the Random Forest (RF) baseline are identical to those reported in Table 4.2 and are included here for ease of comparison

	Yucatan		**Brazil**		**Australia**	
Model	AP Det	mAP Class	AP Det	mAP Class	AP Det	mAP Class
BatDetect2$_{zero}$	0.811	—	0.659	—	0.921	—
BatDetect2$_{tuned}$	0.835	0.407	0.830	0.839	0.940	0.526
BatDetect2$_{full}$	0.923	0.803	0.926	0.962	0.973	0.70
Random Forest	0.649	0.430	0.883	0.912	0.928	0.603

The classification performance, measured by mAP Class, was considerably lower than the detection performance. BatDetect2$_{tuned}$ generally underperformed compared to the Random Forest baseline. The mAP Class for BatDetect$_{tuned}$ was 3-7 percentage points lower than the Random Forest across all datasets. On the Brazil dataset, which has only three sonotypes, BatDetect2$_{tuned}$ achieved an mAP Class of 0.830, while on the Yucatan dataset, performance was considerably low at 0.407.

4.5 Discussion

Here, I have shown that by modifying the base CNN architecture with the addition of a self-attention layer considerably improves classification performance. This improvement is particularly pronounced for species within the *Myotis* genus, suggesting that leveraging the temporal structure of call sequences may be crucial for

accurate classification within this group. Through its self-attention layer, BatDetect2 efficiently utilizes information from longer input timescales without significantly increasing computational cost, resulting in a model that can perform inference approximately 17 times faster than real-time on a GPU. However, modifications aimed at enhancing the use of frequency data did not yield notable performance improvements. Although the proposed modification was relatively simple, and other approaches might enhance performance, it is also possible that a fully CNN model already effectively captures this frequency information. This aligns with findings that CNNs, while translation invariant in theory, often exhibit subtle violations of this property in practice (Zhang, 2019). Overall, these findings highlight the importance of incorporating longer audio context and long-range temporal patterns into the design of deep learning models for bat call classification, and potentially for other bioacoustic tasks. Further exploration of audio-specific architectures, particularly those utilizing raw audio directly instead of 2D image representations (Ravanelli & Bengio, 2018; Hagiwara, 2023), holds significant potential for further performance improvements (Stowell, 2022).

I showed that the full BatDetect2 model is able to learn to detect and classify echolocation calls from bats across five different datasets. BatDetect2 significantly outperforms the traditional call parameter-based baseline, providing a strong argument in favour of Deep Learning (DL) models over traditional methods for this task. Despite the growing trend towards DL methods, direct comparisons with call parameter-based methods remain scarce, for example only in Mac Aodha et al. (2018) for detection and Fundel et al. (2023) for classification. For the majority of species in the challenging UK_{diff} dataset split, BatDetect2 results in high precision at high recall rates (Figure 4.4). This is important as it enables practitioners to trade-off recall for precision to ensure that they obtain reliable, high confidence, predictions from the model. The file-level accuracy is 78% and 86.6% for the UK_{diff} and UK_{same} datasets, where a large percentage of the mistakes can be attributed to known challenging species, i.e. the *Myotis* species. Although no clear relationship between the number of training examples and performance was observed for the

UK_{diff} dataset, the higher performance of the model trained on UK_{same} suggests that larger and more representative training datasets can improve model robustness. However, it is difficult to disentangle the effect of increased training data size from the more rigorous, independent train-test split used for UK_{diff}, where recordings from the same source were not shared between training and testing sets. Further investigation is warranted to clarify these effects.

Using a pre-trained BatDetect2 model as a basis for transfer learning yielded inconsistent performance across different regions and species. The model's ability to detect bat calls, even in diverse environments, was promising, as demonstrated by the high detection metrics (AP Det) in the Yucatan and Australia datasets. However, performance on the Brazil dataset was significantly lower, potentially due to the prevalence of low-frequency calls in that region (Figure C.4). Fine-tuning the model on the target datasets did improve detection performance, suggesting an ability to adapt to new species. In general, the off-the-shelf and fine-tuned BatDetect2 models outperformed the call-parameter baseline in detection, though not on the Brazil dataset. In contrast, the classification performance (mAP Class) of the fine-tuned model ($BatDetect2_{tuned}$) was notably weak across all datasets and was surpassed by the Random Forest baseline in the Yucatan and Australia datasets. This is likely because BatDetect2 extracts only 32 features, compared to 268 used in the Random Forest baseline. Therefore, increasing the number of features used to represent calls might improve transfer learning performance. Potentially, a larger model with more data for pre-training could lead to a model that performs better when transferred to new regions.

BatDetect2 performs well across the five datasets tested, however it still relies on the availability of diverse, and exhaustively annotated, training data. Collecting such data can be challenging, in addition to being time-consuming to annotate, as explored in Chapter 2. While methods for semi-supervised and self-supervised training offer the potential to learn effective models with limited to no training supervision (Heggan et al., 2024; Hagiwara et al., 2023), diverse labelled data is still needed to evaluate the performance of the developed models. Bat calls can exhibit plasticity depending

on the population sampled (Montauban et al., 2021), the presence of other species, and the composition of the local environment. As a result, care needs to be taken to ensure that the collected training datasets are representative of the downstream deployment situations as much as possible (van Merriënboer et al., 2024). Finally, our training datasets currently only contain annotated echolocation calls, and thus the model cannot make predictions for other types of calls, e.g. social calls or feeding buzzes. This limitation could be addressed with appropriate training data.

Unlike typical deep learning-based classifiers, BatDetect2 returns a list of interpretable detections for a given input recording, each represented by a time-frequency bounding box around the detected call and associated species probabilities (see Figure 4.3). This is valuable as it enables easier inspection of the model's predictions, facilitating a better understanding of potential failure cases. However, it is left up to the user to decide how to best merge the individual detections into a set of 'bat passes', where a pass constitutes a sequence of individual calls. This aggregation step is often crucial, as downstream analysis are typically derived from the number of detected individuals or their activity levels (e.g., Ferreira et al. (2022) and Hoggatt et al. (2024)), rather than the number of detected calls. One approach is to use a grouping-based heuristic based on the time between detected calls as in Mac Aodha et al. (2018). The high recall rates of BatDetect2 means that this type approach is less likely to separate individual bat passes into multiple different ones. In contrast, methods that produce high numbers of false negatives run the risk of over-counting the number of passes as they can miss faint calls in a sequence, and thus incorrectly break them up into a number of shorter passes. Still, a better understanding of how to best merge these detections into passes is needed, particularly for distinguishing calls from individual bats. This would allow for more accurate estimates of the number of individuals present, improving the reliability of population monitoring.

In this study I demonstrate that the proposed training pipeline can be applied to audio data from distinct regions without requiring modifications to the underlying code. This pipeline is packaged in the open-source Python package `batdetect2`, available on GitHub. In conjunction with accessible annotation tools like `whombat`

(Chapter 2), `batdetect2` enables the training and deployment of models on custom annotated datasets, even beyond bat species. However, while no coding is required to train a model, some technical expertise is still needed to set up the training environment and to understand the model's output. Integrating the training pipeline into user-friendly, graphical interface tools could offer a more accessible solution for practitioners. Ultimately, this work helps to democratises the development of specialised bioacoustic models by removing significant technical barriers, thereby enabling practitioners to focus on collecting and annotating datasets for their species of interest.

Chapter 5

acoupi: An Open-Source Python Framework for Deploying Bioacoustic AI Models on Edge Devices

5.1 Abstract

Passive Acoustic Monitoring (PAM) coupled with Artificial Intelligence (AI) is becoming an essential tool for long-term biodiversity monitoring across vast landscapes. Traditional PAM systems often require frequent manual data offloading and impose substantial demands on data storage and computing infrastructure. Deploying smart bioacoustic devices that can process and analyse data on-device, transmitting only relevant information, can significantly reduce manual data offloading and the volume of data requiring storage and processing. However, programming these devices for robust operation is challenging, requiring specialised knowledge in embedded systems and software engineering. Despite the growing development of AI models for bioacoustic monitoring, their full potential remains unrealized without accessible tools for deploying them on customised hardware and adapting device behaviour to specific monitoring goals. To address this challenge, I develop `acoupi`, an open-source Python framework that simplifies the creation and deployment of smart bioacoustic devices. `acoupi` integrates audio recording, AI-based data processing,

data management, and real-time wireless messaging into a unified and configurable framework. By modularising key elements of the bioacoustic monitoring workflow, `acoupi` allows users to easily customise, extend, or select specific components to fit their unique monitoring needs. The flexibility of `acoupi` is demonstrated by the integration of two bioacoustic classifiers: BatDetect2, developed in Chapter 4, for UK bat species classification, and BirdNET for bird species classification. I also present a month-long field deployment of two `acoupi`-powered devices in a UK urban park, demonstrating the framework's reliability. `acoupi` can be readily deployed on low-cost, low-power hardware, such as the Raspberry Pi, and its customisable design supports a wide range of monitoring applications. By providing a standardised framework and simplified tools for creating and deploying smart bioacoustic devices, `acoupi` lowers the barrier to entry for researchers and conservationists, facilitating the broader adoption of AI-powered PAM systems.

5.2 Introduction

With the growing need for biodiversity conservation, recovery, and management (IPBES, 2019), it is essential to consider and develop techniques for scaling such efforts efficiently (Besson et al., 2022). Governments worldwide are increasingly committing to biodiversity conservation goals under the Kunming-Montreal Global Biodiversity Framework (CBD, 2022), thereby creating incentives and obligations for the generation of accurate, comprehensive and transparent data on the state of biodiversity (Stephenson et al., 2022). Within this context, Passive Acoustic Monitoring (PAM) has emerged as a key approach for conducting biodiversity assessments and generating broader ecosystem analyses (Browning et al., 2017; Gibb et al., 2018; Ross et al., 2023). The decreasing cost and miniaturisation of acoustic devices, such as the open-source AudioMoth (Hill et al., 2019), have significantly expanded the capacity for deploying extensive monitoring networks of acoustic devices (Sethi et al., 2020). Moreover, the development of Artificial Intelligence (AI) tools for automating the detection of key acoustic signals within the collected data (Kahl et al., 2021; Stowell, 2022; Chapter 4) enables researchers to obtain

evidence of faunal activity from all deployed devices within the network (Sethi et al., 2024). This has facilitated long-term acoustic studies across various scales, from local to continental (Roe et al., 2021), and across diverse environments, from urban (Fairbrass et al., 2019) to remote and challenging locations (Ross et al., 2023), encompassing both terrestrial (Sugai et al., 2018) and marine species (Mooney et al., 2020).

Deploying PAM systems, however, typically requires frequent visits for maintenance tasks on individual devices, including data retrieval, storage media replacement, and battery changes (Browning et al., 2017). While devices like AudioMoth (Hill et al., 2019) or Solo (Whytock & Christie, 2016) offer configurable recording schedules to adjust sampling effort and resource consumption, monthly visits are common (Karlsson et al., 2021), posing logistical challenges for deployments in extensive, fragmented, or remote locations. Furthermore, data must be physically retrieved from SD cards in the field and transported to a central location for analysis (Roe et al., 2021; Karlsson et al., 2021) introducing risks of data loss or corruption (Fig 5.1a). The inherent physical separation between data collection and processing introduces significant delays in inferring ecological insights and hampers the timely detection of time-sensitive events, such as illegal hunting activity.

Modern networking technologies offer the potential to significantly accelerate data transfer from field deployments. The combination of cellular, Wi-Fi, or Long-Range Wide-Area Network (LoRaWAN) communication with continuous power sources such as solar panels enables significantly longer deployments without intervention (Li et al., 2015). Examples include large-scale wildlife monitoring with cellular networks in Borneo (Sethi et al., 2020) and Norway (Bick et al., 2024), and Wi-Fi networks for monitoring dolphins in the Mediterranean (Brunoldi et al., 2016). However, transferring large audio files, particularly high-sample-rate recordings needed to capture ultrasonic vocalisations like bat echolocation calls (Jones & Holderied, 2007) or rat social communication (Coffey et al., 2019), can be challenging due to fluctuating cellular data speeds in areas with suboptimal coverage or the inherent bandwidth limitations of LoRaWAN (Adelantado et al., 2017). Critically, storing and

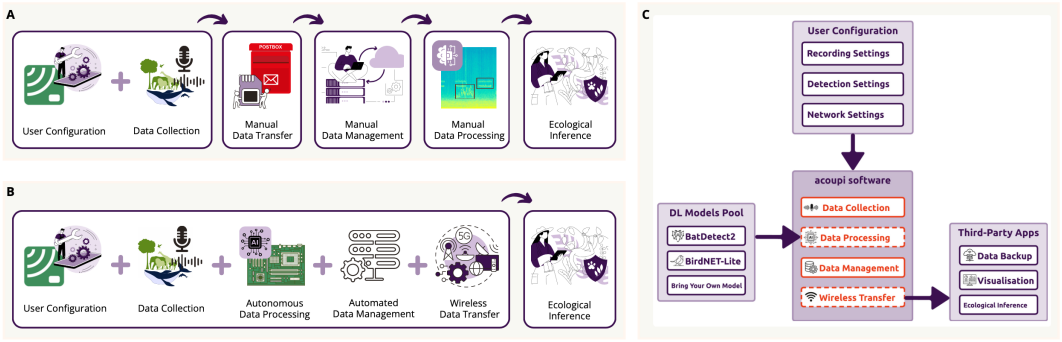


Figure 5.1: Overview of `acoupi` (A) Traditional passive acoustic monitoring workflows consists of fragmented steps requiring frequent intervention, limiting scalability. These steps include device deployment, data retrieval and transfer to a central location, data management, processing with AI models to extract acoustic events, and finally, ecological inference. (B) `acoupi` integrates this workflow into a single device that supports on-board AI classification and wireless data transfer, reducing interventions and accelerating data turnaround. (C) `acoupi` offers a plug-and-play approach that allows users to configure the workflow to their monitoring needs. Users can specify configuration parameters, select a classifier from the AI models pool, and set up wireless network endpoints for integration with third-party applications. `acoupi` coordinates essential tasks (orange) like data collection and management, as well as modules for data processing, transfer and reporting, which are optional (dotted) for added flexibility.

processing the transferred audio data is challenging, as large-scale deployments generate vast quantities of recordings (e.g., tens of millions of hours for the Australian Acoustic Observatory), resulting in significant storage and management costs (Sethi et al., 2018; Roe et al., 2021). Post-deployment processing with AI models requires specialised infrastructure and substantial computing power (Sethi et al., 2020; Stowell, 2022), potentially hindering the adoption of acoustic monitoring by research teams lacking the necessary resources or expertise.

Edge computing (Hua et al., 2023), which involves executing AI models directly on devices deployed in the field, offers a compelling solution to the challenges of post-deployment processing. This approach is increasingly adopted across research and industrial applications to reduce computational burden on centralised infrastructure and enhance system responsiveness (Baucas & Spachos, 2020). Early examples of edge computing applications for biodiversity monitoring include monitoring bats (Zualkernan et al., 2021; Gallacher et al., 2021), birds (McGuire, 2024; Disabato et al., 2021), wolves (Stähli et al., 2022), as well as monitoring urban noise

levels (Baucas & Spachos, 2020; Baucas & Spachos, 2024) and assessing bee-hive health (Chen et al., 2024). The hardware used in these projects fall broadly into two categories: microcontroller units (MCUs) and single-board computers (SBCs). MCUs are power-efficient but have limited computational capacity, which restricts the complexity of implementable AI models and requires proficiency in low-level programming languages for customisation (Disabato et al., 2021). Conversely, SBCs, such as the popular Raspberry Pi (RPi), are versatile and beginner-friendly (Jolles, 2021), integrate with various peripherals and sensors, and support the use of high-level programming languages like Python, a tool increasingly common in ecological research (Lapp et al., 2021; Ulloa et al., 2021; Chapter 2). The BirdNET-Pi project (McGuire, 2024), a popular example, demonstrates the application of RPi-type boards in creating real-time bird monitoring stations, powered by the accessible BirdNET AI model (Kahl et al., 2021). However, existing solutions for edge processing for biodiversity monitoring use a rigid software architecture that is tightly coupled to the specific hardware and AI model, hindering adaptation to evolving hardware and AI models. Furthermore, developing software for edge devices capable of simultaneously coordinating recording, processing, and data communication within a single device presents a significant engineering challenge. While AI models for the detection of bioacoustic signals are rapidly advancing (Höchst et al., 2022), their potential for biodiversity monitoring remains underutilised without accessible mechanisms for integrating these models within configurable edge devices that can adapt to specific project needs.

To fill the gap in accessible tools for bioacoustic edge processing, I developed `acoupi`, an open-source Python framework that simplifies the development and deployment of networked devices for edge processing for bioacoustic. This framework enables the creation of custom programmes for managing the entire bioacoustic workflow, from audio capture and on-device AI-powered processing to data management and wireless transmission (Fig. 5.1b). Key features of `acoupi` include simplified integration of custom AI models, easy fine-tuning of device behaviour through configuration settings, and robust deployment on a range of compatible SBC-

based devices (Fig 5.1c). To demonstrate its capabilities, I integrate two pre-trained bioacoustic classifiers, BatDetect2 (Chapter 4) and BirdNET (Kahl et al., 2021), and evaluate their performance within the `acoupi` framework following a month-long deployment. Finally, I discuss the limitations of `acoupi` and offer key considerations for its effective use

5.3 Software Overview

The `acoupi` software is structured in two main parts: a *framework* that provides tools for building programmes and an *application* that manages the configuration and execution of these programmes on edge devices. Central to `acoupi` is the concept of a *"programme,"* defined as a collection of tasks or routines executed by the device (Figure 5.2). Each task represents an independent unit of work, often running in parallel with other tasks. Tasks can be scheduled, such as periodic recording, or triggered by other tasks, for instance, processing a recording with an AI model upon its completion. The `acoupi` *framework* provides a structured and standardised approach for defining programmes, promoting flexibility of programme design to meet diverse user needs. The `acoupi` *application* ensures the harmonious and fault-tolerant execution of a programme. Moreover, it allows users to customise programme parameters via a simple command-line interface (CLI), facilitating a "no-code" approach. In the subsequent sections, we provide a detailed overview of the `acoupi` framework, followed by a set of requirements to use and run the `acoupi` application.

5.3.1 acoupi Framework

The `acoupi` framework is designed to simplify the creation of customised programmes. While customisability remains the main objective, a key secondary goal is programme standardisation, ensuring all programmes adhere to a consistent structure for defining inputs, behaviours, and outputs. This standardisation offers several key advantages. First, user customisations are guaranteed to function correctly within the `acoupi` framework. Standardised programme structures promote easy sharing and collaboration among users. Consistent inputs and outputs facilitate integra-

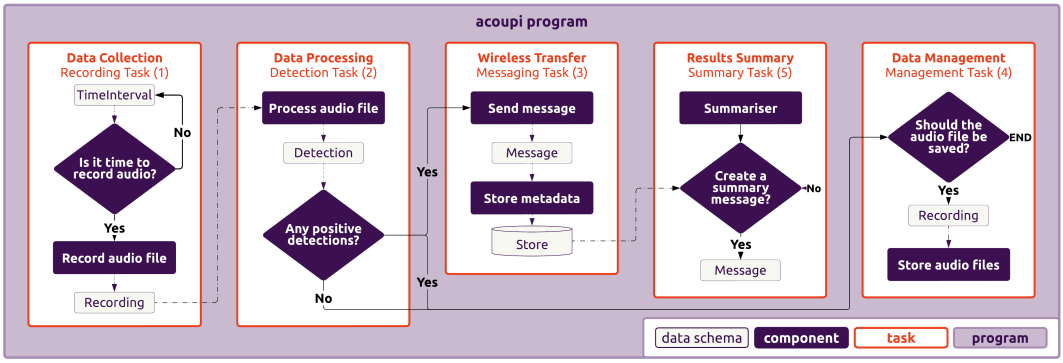


Figure 5.2: Example of a simplified `acoupi` program. This program (mauve) implements five tasks: (1) recording, (2) detection, (3) messaging, (4) management, and (5) summary. Each task (orange) follows a standardised workflow of individual steps (dark purple), involving actions (rectangles) and decisions (rhombuses). These steps are carried out by modular software components. Users can exchange these components to modify device behaviour, customising how actions are performed and decisions are made without altering the overall workflow. Component behaviour can be fine-tuned through user-provided configuration parameters. Standardised data objects (light grey) are passed between components, ensuring consistency across the workflow.

tion with other devices and third-party services. Finally, standardisation within `acoupi` establishes a common language for easily understanding and discussing programme design. This section presents a brief overview of the tools for programme customisation provided by the `acoupi` framework.

Initially, `acoupi` provides *programme templates* that require minimal modification to create fully functional programmes. Each template offers a set of pre-defined tasks that can be readily extended or adapted for more complex applications. For instance, the `DetectionTemplate` includes tasks for recording audio according to a user-specified schedule, processing recordings using an AI model to detect acoustic events of interest, and transmitting detection results to a remote server. To create a programme with this template, the user needs only to provide the specific AI model to be used (Figure 5.1c). To ensure compatibility, `acoupi` defines a standardised input and output format for bioacoustic AI models, and any model adapted to this format can be seamlessly integrated into `acoupi` programmes. `acoupi` offers several such basic templates as starting points for programme creation, enabling users to quickly develop functional programmes for common bioacoustic monitoring scenarios.

For more specialised applications, users can augment the pre-defined templates with custom tasks. While tasks in `acoupi` can be any user-defined Python function, providing developers full control, the framework offers *task templates* to facilitate standardisation and streamline development. These task templates cover common operations, including: (1) *recording*, (2) *detection*, and (3) *messaging*, as mentioned previously, as well as (4) *management* tasks for data storage and file handling, (5) *summary* tasks to generate periodic analytical reports, and (6) *heartbeat* tasks to monitor system health. Each task template utilises a set of user-provided components to execute a predefined workflow (Figure 5.2). For example, a recording task requires a `Recorder` component (responsible for interacting with the microphone) and a storage component to manage the metadata of captured recordings. Similar to its approach with AI models, `acoupi` defines a clear interface for components like `Recorder`, allowing users to integrate diverse recording mechanisms.

`acoupi` provides a collection of modular components that serve as building blocks for constructing tasks. Unlike tasks, which represent complete units of work, components encapsulate specific functionalities within a task (Figure 5.2), such as audio recording (`Recorder`), species detection (pre-adapted AI models), data transmission, or structured data storage. For example, the `Recorder` component simply captures audio for a specified duration, while a recording task might check recording conditions, capture audio, and store associated metadata. This modularity allows components to be reused across different tasks, promoting consistency. All components in `acoupi` adhere to a set of definitions or interfaces, called component types, which clearly define the requirements for building a component of that type, including its functionality, inputs, and outputs. To ensure reliable data exchange between components, `acoupi` utilises standardised data objects. These objects represent the various data types generated and used during programme execution, such as `Recording`, `Detection`, and `Message`. This standardisation ensures data consistency and compatibility throughout the programme. While `acoupi` provides a range of predefined components for immediate use, it also allows users to expand the set of components while ensuring correct integration with the rest of the system.

Programmes in `acoupi` can be configurable, allowing users of the programme to adjust its parameters without modifying the underlying code. This enables a "no-code" adaptation to specific deployment needs, such as modifying the recording schedule and duration to adjust sampling effort, or specifying the address and authentication credentials when transmitting detections to a remote server. `acoupi` requires that all the adjustable parameters of a programme be specified upfront in a configuration schema—a structured blueprint that defines the allowed parameters and their expected format. The configuration schema serves to inform the users about which parameters are adjustable and can be used to validate the provided configuration before deployment. By designing the tasks and configuration schema of a programme, `acoupi` empowers developers to easily create reusable programmes readily adaptable to diverse needs without further coding.

A comprehensive overview of pre-built components, tasks, and programmes is available in the online documentation (https://acoupi.github.io/acoupi/), along with detailed guidance on creating custom programmes at (https://acoupi.github.io/acoupi/howtoguide/programs/).

5.3.2 acoupi Application

The `acoupi` application enables the execution of a pre-built programme on a chosen edge device (Figure 5.3). The application provides a command-line interface (CLI) with simple commands to manage and deploy programmes. The command *"acoupi setup"* guides users through a configuration wizard, allowing them to select a programme and configure its parameters. The validity of configurations can be checked and modified at all times using the command *"acoupi config "*. Once a programme is configured, users can initiate deployment with the command *"acoupi deployment start"*. The application performs pre-deployment health checks to verify the programme configuration and system setup, identifying potential issues such as connectivity problems or microphone malfunctions. Finally, the command *"acoupi deployment stop"* shuts down the system and records the start and end times of a deployment to track monitoring effort.

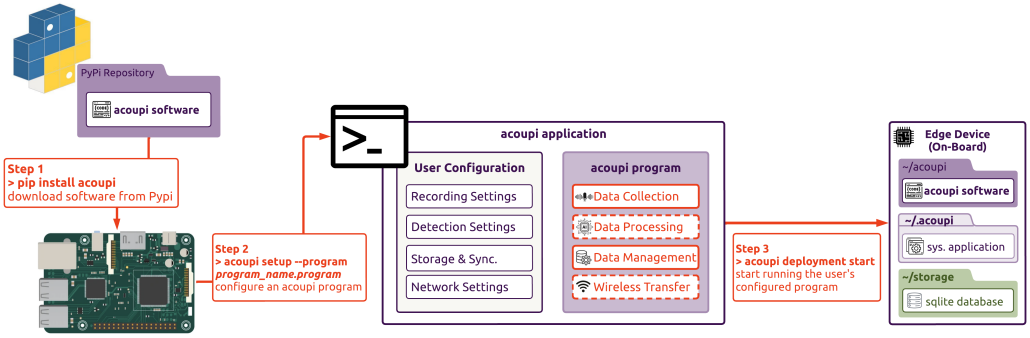


Figure 5.3: Overview of an `acoupi` application deployment process. (1) Download and install the `acoupi` software from the PyPI repository onto the target device. (2) Use the command line interface (CLI) to configure the program. The CLI prompts the user for parameters to configure recording, processing, data management, and messaging tasks. (3) Initiate deployment through the CLI. This triggers health checks to ensure the system is configured correctly and upon completion begins executing tasks according to the defined schedule.

The `acoupi` application ensures the timely execution of programme tasks, even in challenging conditions. Maintaining reliable operation on edge devices can be difficult due to computational resource limitations, network instability, and power fluctuations. To address this, `acoupi` leverages Celery (Solem & Saif Uddin, 2024), a robust and widely-used task management tool. Celery helps coordinate and schedule tasks, automatically retry them if they fail, and run multiple tasks simultaneously whenever possible. Furthermore, `acoupi` incorporates mechanisms for automatic recovery after power failures. A comprehensive log of device activities is maintained, aiding in identifying failures and preventing data loss.

To optimise storage usage during long deployments, `acoupi` does not store recordings by default. Instead, recordings are held temporarily in the working memory for processing. Depending on the programme's logic and configuration, recordings may be selectively saved to disk, such as when an AI model identifies vocalisations of target species. However, both the chosen programme and its configurations are stored, facilitating the reproducibility of deployments by sharing configuration files. Additionally, `acoupi` stores lightweight SQLite databases containing essential recording metadata and automated detections. This design helps mitigate the risk of premature deployment termination due to storage depletion while maintaining

crucial metadata for subsequent analysis and reproducibility.

Documentation for programme deployment is detailed at https://acoupi.github.io/acoupi/tutorials/configuration/.

5.3.3 Requirements

To run `acoupi`, a single-board computer (SBC) running a Linux Operating System (OS) is required. `acoupi` has been extensively tested on a Raspberry Pi 4 Model B running the 64-bit Raspberry Pi OS; however, devices with similar specifications should also be compatible. Raspberry Pi systems are especially recommended for new users due to their ease of use and extensive documentation (Jolles, 2021).

In addition to the computing board, a microphone and a microSD card are required. Microphone selection should consider the desired sampling rate, which depends on the target species' vocalisation frequencies. To ensure adequate capture, the sampling rate should be greater than twice the highest frequency of the target species. A microSD card with a minimum capacity of 32 GB is recommended. Users should select a larger capacity microSD card or consider using an external hard drive, according to the volume of audio files they wish to archive for offline analysis post-deployment.

The `acoupi` software is freely available through the Operating System (PyPI) or by downloading it from the GitHub repository https://github.com/acoupi/acoupi. A detailed step-by-step installation guide can be found at acoupi.github.io/acoupi/#installation.

5.4 Pre-Built Bioacoustic Programs

In addition to the `acoupi` framework and application, I have developed two ready-to-use programmes: *acoupi_birdnet* and *acoupi_batdetect2*. These programmes offer out-of-the-box functionality that can be customised through configuration adjustments, without requiring any coding. Both programmes are built upon the `DetectionTemplate` described in the previous section and thus inherit a common structure while incorporating distinct AI models and default configurations.

These programmes leverage two established AI models for the acoustic detection of birds and bats. *acoupi_birdnet* employs the BirdNET model version 2.4 (Kahl et al., 2021), capable of detecting approximately 6,400 bird species globally, along with other relevant acoustic events such as frog calls, insect sounds, domestic animals, fireworks, and engine noise. *acoupi_batdetect2* utilises the BatDetect2 model developed in Chapter 4, designed to detect echolocation calls from 17 bat species commonly found in the UK. BirdNET and BatDetect2 models were trained using recordings made at 48kHz and 256kHz respectively, thus using the same or similar sampling rate when using this models is recommended for better performance. These models have shown good performance within the scope of their original evaluations, but it is essential to acknowledge their potential limitations in broader applications (Pérez-Granados, 2023). As with all AI models, there is potential for misidentification or missed detections, particularly in environments that diverge significantly from the training data (van Merriënboer et al., 2024). Thorough evaluation of model performance within the specific deployment context is strongly recommended (Wood & Kahl, 2024).

Both *acoupi_batdetect2* and *acoupi_birdnet* feature automated and scheduled recording, processing with their respective AI models, and transmission of detections to a remote server. *acoupi_batdetect2* records three seconds of audio every ten seconds between 19:00 and 07:00, while *acoupi_birdnet* captures nine second at the same frequency between 03:00 and 23:00. Detections exceeding a predefined confidence threshold are transmitted to a remote server every 30 seconds. A heartbeat signal is transmitted every 30 minutes to monitor device health, regardless of recording activity or detection events. The programmes allow for optional storage of recordings with confident detections, facilitating post-deployment validation. Crucially, all operational parameters, including recording schedules, durations, frequencies, and messaging intervals, are fully configurable. It is essential to carefully consider the monitoring goals and adjust these settings accordingly (Teixeira et al., 2024).

5.5 Software Testing

To test the reliability of `acoupi`, I configured and deployed both the *acoupi_batdetect2* and *acoupi_birdnet* programmes on two separate Raspberry Pi 4 Model B (RPi) devices. The RPis were deployed at the People and Nature Garden Lab in One Pool Street within the Queen Elizabeth Olympic Park in London, UK (Fig 5.4). This location was selected for initial software testing due to the convenient access to power and a Wi-Fi network, acknowledging that such conditions may not fully represent the challenges of field deployments. Most of the configuration parameters for the *acoupi_batdetect2* and *acoupi_birdnet* programmes followed the default configuration (see Table D.1 for full settings). The devices were deployed for 30 days between October and November 2024.

To evaluate the reliability of the software, I examined key metrics, including recording consistency, processing success, and message delivery (Table 5.1). Both programmes successfully recorded at every scheduled interval; however, minor imprecisions in the scheduler resulted in an average recording frequency of approximately 10.005 seconds. The *acoupi_birdnet* programme successfully processed all but four recordings made shortly after deployment. The *acoupi_batdetect2* programme exhibited a similar success rate, processing 98.2% of recordings. On average, *acoupi_birdnet* took 1.2 seconds to process each 9-second audio clip, while *acoupi_batdetect2* took 5.8 seconds to process each 3-second clip. Both programmes successfully delivered all generated messages, demonstrating reliable message delivery under good network conditions. Despite these positive results, both deployments encountered premature termination. The device running the *acoupi_birdnet* programme was likely dislodged by strong winds, resulting in power loss. The *acoupi_batdetect2* programme encountered a software issue that prevented the processing of 1.8% of the recordings, but this issue has since been resolved.

While not the primary focus of this study, the detections made by the two bioacoustic classifiers were consistent with the expected soundscape of the deployment location and the seasonality of the test. The roof garden is urban, close to a busy traffic

Figure 5.4: Deployment of `acoupi` devices at the People and Nature Lab Garden (A) Sensing Unit Components: single-board computer (Raspberry Pi 4B) and ultrasonic microphone (Dodotronic Ultramic 250k). (B) Protective Enclosure: housing two SBCs, one running `acoupi_batdetect2` and the other `acoupi_birdnet`. (C) Field Deployment: enclosure mounted alongside other environmental sensors at the UCL East campus, Stratford, London.

road, a railway track, but in the proximity of the Waterworks River, where common water birds are found. Common UK bird species identified with high confidence (score > 0.85) included the Eurasian magpie (n=308), Eurasian wren (n=61), Redwing (n=50), European robin (n=38), White wagtail (n=31), Broad-winged hawk (n=25) and European herring gull (n=18). As expected, anthropogenic sounds were prevalent, with engine noise (n=73) and sirens (n=273) being the most frequently detected, followed by fireworks (n=175), likely associated with festivities during the deployment period. The *acoupi_batdetect2* programme did not detect any bat echolocation calls with high confidence (scores > 0.85), and the 174 pulses detected with moderate confidence (scores > 0.4) are likely false positives. This low number of bat detections is consistent with the deployment period (November), when most

Table 5.1: Reliability metrics for deployment of *acoupi_birdnet* and *acoupi_batdetect2* programmes. The metrics presented comprise the total audio recordings captured, the number and percentage of recordings not processed by the AI models, and the number and percentage of messages successfully delivered to the remote server.

		Processing Failures		Messages Sent	
Programme	**Recordings**	Count	% Total	Count	% Success
acoupi_birdnet	129,939	4	0.003%	8716	100%
acoupi_batdetect2	65,711	1,203	1.83%	868	100%

bats in the UK are hibernating. Detections were not validated post deployment(see Appendix D.2 for a detailed summary of the detections made by the two bioacoustic classifiers). Importantly, the absence of bat detections reflects external factors influencing detectability, such as seasonality and the specific AI model used, rather than acoupi's performance. The main goal of this test deployment was to assess the ability of acoupi to reliably execute all scheduled tasks, and testing the ability to detect the target species was outside the scope of this test.

5.6 Discussion

Here I have shown how `acoupi` can be used to embed two bioacoustic AI models, BirdNET and BatDetect2 (developed in Chapter 4), on edge devices. While BirdNET covers a wide range of avian species (Kahl et al., 2021) and BatDetect2 targets all bat species found in the UK, integrating additional bioacoustic AI models will be necessary to accommodate a greater diversity of species and applications. Although most current AI models, including BirdNET and BatDetect2, are based on Deep Learning (DL), a specific subclass of AI, acoupi can theoretically integrate any AI model. This even includes non-Machine Learning models like the toolbox for animal detection Tadarida (Bas et al., 2017) and the frog detector RIBBIT (Lapp et al., 2021). However, model integration requires considering their size and complexity, as these factors directly impact processing speed and power consumption on edge devices (Desislavov et al., 2023). If processing times exceed the recording interval, audio backlogs and potential system overloads can occur. Among AI

models, DL-based models are often computationally demanding, but techniques like quantisation (Rokh et al., 2023), pruning (Cheng et al., 2024), and knowledge distillation (Gou et al., 2021) can reduce their size and complexity. A trade-off exists between model size and detection performance, making it essential to evaluate the impact of optimisation techniques on detection accuracy (Desislavov et al., 2023). Future work should focus on optimising bioacoustic models, such as BatDetect2, for edge deployments, investigating which techniques enable optimal compression while retaining good performance on common bioacoustic tasks. As models designed for edge processing in bioacoustics emerge (for example see Höchst et al. (2022), Disabato et al. (2021), Zualkernan et al. (2021) and Ghani et al. (2023)), `acoupi` will serve as a platform for integrating these models and making them accessible to the community.

In its month-long deployment at the People and Nature Lab in London, UK, `acoupi` successfully coordinated audio recording, processing, and transmission, capturing all scheduled recordings, sending all detection messages, and processing the vast majority of recordings with the AI models. Nonetheless, premature termination in both deployments underscores the need for more extensive field-testing, particularly in less favourable conditions, to assess `acoupi`'s real-world robustness. While `acoupi` leverages widely used software tools for ensuring reliable operation, empirical evaluation of its robustness in an experimental setting with varying network connectivity and power availability is crucial. Furthermore, although `acoupi` is designed to run on any Linux-compatible single-board computer (SBC) further testing across a variety of SBCs could provide valuable insights into the software's compatibility and performance across different hardware platforms. Addressing these challenges will be a focus of future `acoupi` development, with iterative improvements informed by further field-testing. The codebase includes automated testing to facilitate modifications and a system for distributing updates, which can be applied remotely, supporting the system's maintenance and long-term adaptability.

In its current state `acoupi` is limited to audio recording as the main data collection method, and Wi-Fi as the main communication channel. However, SBCs like the

Raspberry Pi offer versatile options for integrating additional sensors, extending data storage, and enabling alternative connectivity methods like cellular and LoRaWAN (Jolles, 2021). One promising extension is the integration of multichannel audio recorders (Heath et al., 2024), enabling on-board localisation algorithms to estimate the position of vocalising animals, a crucial step towards more accurate population density estimations (Rhinehart et al., 2020). Integrating additional sensors to capture abiotic data, such as rainfall, wind speed, humidity, and temperature, could provide crucial context for ecological analyses, as these factors directly affect sound transmission and can mask relevant target sounds affecting detectability (Metcalf et al., 2023; Ross et al., 2021). Moreover, an `acoupi` deployment could be expanded with low-cost camera modules to create smart camera trap systems (Darras et al., 2024), utilising the existing hardware infrastructure and `acoupi`'s capabilities for data integration and processing. While not yet available in `acoupi`, the modularity of the framework and its open-source nature provide a foundation for the community to integrate these extensions and contribute to its ongoing development.

This work demonstrates that `acoupi` can serve as a flexible framework for deploying bioacoustic AI models on edge devices. However, it is important to acknowledge that `acoupi`'s requirements may limit its applicability to certain monitoring scenarios. Firstly, `acoupi` is designed for SBCs, which typically have higher power consumption than microcontroller units (MCUs) such as the AudioMoth (Hill et al., 2018). Consequently, `acoupi` requires a continuous power source, such as a solar panel with a battery or a direct connection to the mains power, as in the deployments presented here, to ensure uninterrupted operation. Additionally, a complete setup requires a microphone, an enclosure, and any additional sensors, increasing the cost and complexity of the deployment. Although adding a solar panel, battery, and a high-end microphone might increase the cost for budget-constrained projects, the base cost of an `acoupi`-compatible SBC without these additions is comparable to that of an AudioMoth (for a detailed cost breakdown of an analogous complete system see Sethi et al., 2018). For resource-constrained monitoring scenarios low-power or even battery-free devices (Lostanlen et al., 2021) may be

more suitable. In contrast, `acoupi` is ideal for long-term deployments requiring continuous monitoring on key sites, where access to power and network connectivity allows for minimal intervention. Future iterations, however, could incorporate power-management mechanisms (Balle et al., 2024), including intelligent scheduling to selectively power the device, optimising detection probability while minimising power consumption (Millar et al., 2024). Fundamentally, `acoupi` is a software solution and does not mandate a specific hardware setup, thus providing flexibility in the choice of hardware components. Further research into specific hardware recommendations for bioacoustic monitoring, tailored to different project needs, would be a valuable (see for example Darras et al., 2021; Lapp et al., 2023; Metcalf et al., 2023).

Ultimately, `acoupi` aims to provide a flexible and user-friendly tool for bioacoustic monitoring, adaptable to a wide range of monitoring scenarios. For example, `acoupi` deployments can generate detections that could be integrated into live dashboards for near-real-time monitoring of time-sensitive events. This could include mitigating human-wildlife conflicts (Richardson et al., 2020), managing the spread of vocalising invasive species (Wood et al., 2024), or enabling better coexistence between humans and wildlife in urban areas, such as by dimming city lights in response to migratory bird movements (Horton et al., 2019). Detections can also inform ecological research through methods like occupancy modelling (Rhinehart et al., 2022) or call density analysis (Navine et al., 2024). However, validating detections is crucial when using AI models in novel environments (Pérez-Granados, 2023; van Merriënboer et al., 2024), and `acoupi` can facilitate this by storing recordings selected according to specific criteria for later validation. While simple criteria for saving recordings, such as exceeding a detection score threshold, are currently implemented, more sophisticated criteria aligned with specific modelling requirements can be added (Navine et al., 2024; Knight et al., 2020). Moreover, by establishing a common set of standardised concepts, each with specified metadata, `acoupi` facilitates the planning of monitoring surveys in a way that promotes integration and comparability of results across research projects (Besson et al., 2022).

`acoupi` aims to provide greater accessibility to bioacoustic monitoring, empowering interested parties to use passive acoustic monitoring technologies to address ecological questions and contribute to conservation efforts.

Chapter 6

Discussion

Despite past coordinated efforts like the Aichi Biodiversity Targets falling short of their goals (Xu et al., 2021), there is renewed optimism fuelled by increased international commitments under the Kunming-Montreal Global Biodiversity Framework (KM GBF). This framework has brought acoustic monitoring to the forefront as a key component for tracking progress, due to its potential for Scalable, Accessible, Granular, Evidenceable, and Direct (SAGED) metrics (Ford et al., 2024). However, the field faces challenges related to data scarcity and the limited availability of readily usable or customisable tools, leading to a restriction in the effective use of AI-driven acoustic monitoring to specific geographic regions and well-resourced organisations. Without concerted efforts to expand its scope and provide wider access, there is a risk of hampering conservation efforts in critical areas while exacerbating existing biases in data collection and practice, potentially replicating historical injustices driven by unequal access to resources and decision-making power in conservation (Pritchard et al., 2022). In this Thesis, I contributed to overcoming these challenges by presenting novel tools and methodologies designed to accelerate the development and application of AI models for bioacoustic monitoring at scale, ultimately aiming to make this technology a globally accessible tool for biodiversity conservation. The remainder of this chapter outlines the specific contributions of this thesis to the field of AI for bioacoustics, followed by a discussion of key takeaways, limitations, and remaining gaps in the research.

6.1 Summary of contributions

Chapter 2 addresses the need for improved tools to support the annotation of bioacoustic data for AI model development. Acknowledging that, in practice, data preparation and refinement constitute a significant portion of AI development efforts and that data quality can significantly impact model performance (Sambasivan et al., 2021; Roscher et al., 2024), this chapter identifies a key gap: the lack of software tools specifically designed to support the iterative process of annotating audio data for training AI models. Through a review of bioacoustic software tools, I established that existing annotation software does not adequately address the unique requirements of bioacoustic data annotation for AI, particularly in facilitating the feedback loop between annotation, model training, and model evaluation. To address this limitation, I developed `whombat`, a novel, open-source software tool designed to streamline the bioacoustic data annotation process for AI applications. `whombat` offers a user-friendly interface and a suite of features designed to support the annotation workflow, including project management and tracking capabilities, support for flexible annotation with customisable tags, functionality to export annotations in AI-ready formats, and the capability to import model predictions for iterative refinement. I demonstrate the utility and flexibility of `whombat` through two distinct case studies: facilitating the annotation of bat echolocation calls for the Bat Conservation Trust (BCT) in the UK and supporting the annotation of bird vocalisations for researchers in the Pacific Northwest. These case studies provide evidence that `whombat` enables efficient and effective data annotation, and that the resulting annotations contribute to the development of robust and accurate AI models for bioacoustic monitoring such as the one presented in Chapter 4. Ultimately, I contribute a valuable tool that enables researchers and practitioners to create higher-quality annotated datasets.

Building upon the foundation of improved data annotation in Chapter 2, I delve into the question of how to annotate for improving AI model performance in Chapter 3. Expanding upon the findings of (Hershey et al., 2021), which highlighted the benefits of strong annotations, I investigate deeper by quantifying the impact of both the level of detail in spectro-temporal annotations and the amount of training

data on model accuracy. Using a diverse dataset of bat echolocation calls, I trained detection and classification models under various data availability scenarios and annotation approaches. Similar to Morfi & Stowell (2018), I demonstrate that augmenting model training with a localisation task, informed by detailed annotations, enhances classification performance significantly. Notably, in low-data scenarios with only 10 recordings per species, this approach yielded up to a 10% increase in classification accuracy — a gain surpassing that achieved by collecting five additional recordings per species. This finding provides a valuable alternative for enhancing model performance when further data collection is challenging or resource-intensive. While various methods for generating detailed annotations were explored, the research concludes that the mere presence of detailed spectro-temporal information is more critical than the specific creation method. Therefore, I argue for the adoption of bounding boxes as an effective and practical means of incorporating such detailed information into the annotation process. This approach offers a readily implementable strategy to develop more robust and accurate bioacoustic models, even when faced with limited data.

In Chapter 4, I shift the focus to model architecture design as a means of improving performance, introducing a novel architecture that leverages detailed box annotations for joint detection and classification of bat echolocation calls. Instead of relying on existing models, I introduce a modified neural network architecture incorporating fundamental yet simple acoustic principles relevant to bat echolocation. This design, validated on four diverse bat call datasets, yields significant performance gains in both detection and classification tasks compared to traditional parameter-based methods. Furthermore, aligning with the findings of Chapter 3, the model leverages bounding-box annotations to produce more interpretable and ecologically relevant predictions, particularly valuable in real-world scenarios where multiple species co-occur (van Merriënboer et al., 2024). A key innovation of this architecture is its use of a self-attention module (Vaswani, 2017), enabling efficient long-range temporal reasoning. This allows the model to effectively incorporate important discriminative features like inter-pulse intervals, which are important features for distinguishing bat

species (Szewczak, 2004) but are challenging for traditional convolutional networks to capture without increased model complexity. The reduced complexity of the model architecture, combined with the strong supervision provided by the detailed annotations, facilitates model training even with limited data. In this chapter, I not only demonstrate the performance gains of this approach, but it also suggests its broad applicability across diverse geographical regions, offering an adaptable pipeline for automated analysis of bats. While not tested on other taxa, the principles and methods employed are generic, suggesting potential for broader bioacoustic applications. To foster wider adoption, I incorporate these methods into `batdetect2`, an open-source software package that provides access to both the trained model and the training process.

While previous chapters focused on model development, in Chapter 5, I address the need for practical deployment of these models in real-world bioacoustic monitoring. Recognising that traditional post-processing approaches often lead to data management challenges and delayed insights, in this chapter I develop `acoupi`, an open-source software framework designed for acoustic analysis on edge devices. `acoupi` enables researchers and practitioners to create monitoring programmes that integrate audio data collection, AI processing with their model of choice, and data transfer. By supporting near real-time detection of target acoustic events, and offering the ability to tailor programmes with custom schedules and hardware, `acoupi` provides a flexible solution adaptable to diverse monitoring needs. Importantly, although creating a program with `acoupi` requires some initial Python programming, these programmes can be readily shared and deployed by others without any further coding, potentially lowering the barrier to entry for using AI in bioacoustics. This accessibility is demonstrated through two readily deployable example programmes—incorporating the BatDetect2 model from Chapter 4 and the widely-used `BirdNet` model—which I validated during a month-long field deployment in an urban setting in London, UK. Ultimately, in Chapter 5, I contribute an open-source and standardised framework that supports collaboration and further development of customised acoustic monitoring systems, and provides the wider community with a tool to build

upon, refine, and share programmes and components.

6.2 Key takeaways

In this thesis I have explored various aspects of applying AI to bioacoustic monitoring, from data annotation to model deployment. Several key takeaways emerge from this work, highlighting the importance of data quality, domain expertise, and accessible technology in realising the full potential of AI for biodiversity research and conservation.

6.2.1 The critical role of detailed data annotation in bioacoustic AI

A recurring theme throughout my research in the thesis is the critical role of data annotation in successful AI applications. I emphasise in Chapter 2, that data preparation and refinement are not merely preliminary steps but constitute a significant portion of the AI development process, a finding echoed in existing literature (Sambasivan et al., 2021; Roscher et al., 2024). The quality of the data directly impacts model performance (Zha, Bhat, Lai, Yang & Hu, 2023), underscoring the need for tools and strategies that facilitate rigorous and effective data handling. Given the challenges inherent in acquiring large-scale datasets of bat echolocation calls or other key bioacoustic signals, I underscore the importance of prioritising high-quality, detailed annotation as a key strategy for advancing research in this field.

Iterative data annotation plays a critical role in developing accurate and reliable models for bat echolocation analysis, as demonstrated throughout this research. In Chapter 2, I establish the importance of iterative workflows in AI-driven projects, highlighting that they are the norm rather than the exception, and introduces a collaborative annotation platform designed to facilitate this process. In Chapters 3 and 4, I further exemplify the benefits of iterative data annotation in bat call analysis. The datasets used in these chapters involved multiple rounds of annotation, integrating contributions from multiple annotators and several iterations of reviewing and refining the annotations to improve consistency. These chapters also demonstrate

that model performance improves with increased dataset size, highlighting the ongoing need for high-quality annotated data, especially for underrepresented species. The iterative cycle of model improvement and data annotation is recognised by organisations like the Bat Conservation Trust and aligns with the experiences of other researchers (Roscher et al., 2024), reinforcing the broader understanding that iterative workflows are essential for advancing bioacoustics research.

Additionally, my research demonstrates that using detailed spectro-temporal annotations can significantly improve bat echolocation call detection and classification, particularly in data-limited contexts. Chapter 3 provides evidence that such detailed annotations significantly improve performance compared to coarser ones when training data is scarce. The use detailed annotations enables models to learn more efficiently from limited data, as demonstrated by the robust results achieved with smaller datasets in Chapter 4. These findings align with other works in bioacoustics (Hershey et al., 2021; Chasmai et al., 2024), which emphasise the value of detailed annotation for model performance.

Beyond performance gains, detailed annotations also enhance the interpretability of model outputs. Specifically, in Chapter 4, I demonstrate how training models on detailed annotations enables them to precisely localise detected calls within a spectrogram by generating bounding boxes. This capability provides a more granular understanding of call structure and temporal patterns compared to models employing sliding-window analysis. The fine-grained outputs, coupled with the visualisation and analysis tools like `whombat` presented in Chapter 2, facilitate a more thorough review of model predictions. The ability to compare the predicted bounding box with the corresponding annotations within the spectrogram, both visually and quantitatively, allows for more precise assessment of detection and classification performance (Mesaros et al., 2021; van Merriënboer et al., 2024). For example, a tight alignment between a predicted bounding box and the true acoustic event provides compelling evidence that the model is learning the relevant acoustic features rather than relying on incidental correlations of background noise—a claim harder to defend with less precise detections. This improved interpretability

fosters greater confidence in model predictions and can facilitate a more detailed understanding of bat echolocation.

6.2.2 Integrating bioacoustic knowledge enhances AI model performance

This thesis demonstrates that incorporating bioacoustic principles and expertise into the design and development of AI models can enhance their performance and applicability in bioacoustic monitoring. This is evidenced by the gains achieved by the bioacoustically-inspired model architecture introduced in Chapter 4.

In the broader context of AI, improvements in model performance have primarily been driven primarily by increases in model capacity and the accompanying need for increasingly large datasets (Kaplan et al., 2020). However, such data-intensive approaches do not always align with the realities of bioacoustic research, where large, comprehensively annotated datasets are often scarce (Nolasco, Singh et al., 2023). As an alternative, this research demonstrates that incorporating bioacoustic principles into model design can yield significant performance gains even with limited training data. Chapter 4 exemplifies this approach, showcasing how a model architecture informed by bioacoustic principles can effectively learn temporal relationships relevant to bat echolocation classification. Similarly, in Chapter 3, I demonstrate that modifying the training procedure to explicitly encourage the model to learn the spectro-temporal location of each call, a fundamental aspect of bioacoustic signal analysis, can provide performance gains. These examples highlight the benefit of incorporating domain-specific knowledge into both model architecture and the training process.

The development of high-quality datasets for bioacoustic analysis is fundamentally dependent on the expertise of bioacousticians, particularly for the meticulous task of data annotation. While it is sometimes argued that automation reduces the need for expert input, I contend that AI models, on the contrary, amplify the value and reach of expert knowledge. Chapter 2 details the intricacies of annotation, demonstrating how the process relies on the domain expertise of annotators

to accurately identify and classify nuanced acoustic events through the visualisation, interpretation, and aural analysis of audio signals (Fraser, 2018). Expert field bat ecologists, whose nuanced understanding of bat echolocation calls was crucial for creating the high-quality datasets used in this research, generated the detailed spectro-temporal annotations used in Chapters 3 and 4. Without this expertise, the trustworthiness of model outputs would be significantly diminished. To make expert acoustic identification more accessible and facilitate its broader dissemination, I developed user-friendly training workflows for novice bioacousticians within the whombat annotation tool.

6.2.3 Democratising access to AI for bioacoustic monitoring

A key contribution of my research in this thesis is the push towards democratisation of access to AI technology for the wider bioacoustic community. Recognising that the development and application of AI tools for acoustic monitoring involve multiple steps, and that interest in using these tools extends beyond those with specialised technical skills, I emphasise the importance of enhancing accessibility at each stage of the process. By providing tools and resources that cater to users with varying levels of technical proficiency, I aim to promote broader adoption of AI methodologies and allow the bioacoustic community to take ownership of these technologies.

To maximise accessibility, all steps within the AI workflow should be supported by software tools designed for users with diverse technical backgrounds. For example, in Chapter 2, I developed `whombat`, a user-friendly annotation platform that empowers individuals with basic computer literacy to contribute to the creation of high-quality datasets. Beyond basic annotation, `whombat` caters to technically advanced users by providing tools for data preparation and preprocessing for model training. Building on the work in Chapter 4, I developed `batdetect2`, a software tool for automated bat call analysis and custom model training. `batdetect2` offers both an intuitive interface for basic inference and a flexible architecture for advanced customisation by users with greater programming expertise. To facilitate edge computing applications, in Chapter 5 I developed `acoupi`, a software tool enabling users with basic Python

knowledge to design and implement custom programs for edge devices. Importantly, `acoupi` features a code-free deployment mechanism, further democratising the implementation of custom solutions by lowering the barrier to entry.

Beyond mere access, empowering users of AI tools involves enabling customisation for specific needs and facilitating integration with other tools and workflows. Given the diverse applications and inherent variability within acoustic monitoring (Teixeira et al., 2024), adaptable software tools are essential. `whombat` exemplifies this by allowing flexible annotation and tagging, enabling teams to tailor their efforts to specific project needs and evolving methodologies. Similarly, `acoupi` (Chapter 5) allows users to customise data collection and processing regimes, including model selection, recording schedules, and hardware configurations. However, this flexibility is carefully balanced with mechanisms that promote standardisation, such as the adoption of standardised data structures and support for common ontologies where applicable in both tools. This approach ensures that while users can adapt tools to their specific needs, data remains comparable and interoperable across different projects.

6.3 Limitations and future work

A key constraint encountered in this research is the scarcity of comprehensively annotated datasets for bioacoustics (Stowell, 2022). This limitation directly constrained the scope of the studies presented in this thesis. For instance, in Chapter 3, despite analysing the most extensive dataset available for Mexican bat echolocation calls with 101 species (Zamora-Gutierrez et al., 2020), the study was limited to 17 species to meet the experimental design requirement of at least 30 recordings per species. Similarly, in Chapter 4, while all 17 breeding UK bat species were included, limited sample sizes likely impacted the performance of the classification models for some species. The applicability of the findings presented in those chapters is thus largely restricted to the species and recording conditions represented in the datasets used. This highlights the need for more diverse datasets to test the generalisability of results in future bioacoustic research, particularly for model development. Although

recent initiatives have led to the emergence of more diverse bioacoustic datasets designed for model development and benchmarking (Rauch et al., 2024; Chasmai et al., 2024; Hagiwara et al., 2023; Hamer et al., 2023), often derived from citizen science platforms like xeno-canto (Vellinga & Planque, 2015) and iNaturalist (Matheson, 2014), the majority of these data remain unannotated and exhibit a strong taxonomic bias towards avian species. Addressing these gaps, particularly for underrepresented taxa like bats, will require substantial resources for both data collection and annotation (Chasmai et al., 2024). The wider scientific community likely holds a significant volume of potentially valuable recordings for bioacoustic model development, but the perceived lack of direct benefits for data contributors, along with concerns about inadequate attribution or data misuse, discourages data deposition and sharing (Baker & Vincent, 2019; Gomes et al., 2022). This issue is further exacerbated by the fragmentation of existing datasets across multiple repositories, often stored in heterogeneous formats incompatible with modern AI development pipelines. However, the increasing adoption of open data principles in scientific research (Tenopir et al., 2020), and initiatives like Findable, Accessible, Interoperable, Reusable (FAIR) data (Wilkinson et al., 2016) may gradually mitigate these concerns in the future. Still, there is a pressing need to promote discussions and adoption of data standards (e.g., Roch et al., 2016; Wieczorek et al., 2012; TDWG, 2023; Akhtar et al., 2024), fostering greater harmonisation of existing data and encouraging a more open and collaborative approach to data sharing within the bioacoustic research community. Beyond data sharing, another crucial aspect is improving the efficiency of the annotation process itself. To reduce the time and effort required for annotation, future work should focus on improving the efficiency of identifying and annotating relevant sound events within large audio datasets. Active learning and other agile development methodologies offer considerable promise for improving annotation efficiency (Martinsson et al., 2024; Stretcu et al., 2023; Wang et al., 2022; Kath et al., 2024). Tools like `whombat`, developed as part of this thesis, could help to crowdsource the annotation effort, and future work could focus on incorporating more sophisticated annotation workflows.

A significant challenge faced by current bioacoustic models, including those developed in this thesis, is their limited or unknown transferability to real-world passive acoustic monitoring scenarios. Bioacoustic recordings used for model training are often collected using a targeted or focal approach to facilitate the accurate identification of individual subjects. This approach typically involves using specialised audio equipment to obtain isolated, high-quality recordings of the target species, as is common for avian recordings (Kahl et al., 2021), or employing capture-and-release methods, as frequently seen in bat research (Zamora-Gutierrez et al., 2021). For example, the bat classifiers and detectors presented in Chapters 3 were developed using data predominantly acquired through a capture-and-release approach. However, these controlled recording conditions typically differ substantially from those encountered in passive acoustic monitoring settings, where target sounds may be faint or obscured by co-occurring sounds (van Merriënboer et al., 2024). In such real-world settings, model performance drops across all tested cases due to the mismatch between training and deployment conditions (Hamer et al., 2023; Sharma et al., 2022); the specific transferability of BatDetect2, developed in this thesis, however, remains unknown. Because target sounds in passive acoustic monitoring settings are typically embedded within a complex mixture of background sounds, influenced by the broader environmental conditions of the recording site (Pijanowski et al., 2011), it is crucial to re-evaluate model performance whenever these environmental conditions change to ensure continued applicability (Pérez-Granados, 2023). Nevertheless, further research is needed to develop a more comprehensive understanding of how model performance is affected by these variable environmental conditions, coupled with efficient methods for performance re-evaluation under novel scenarios (Knight et al., 2020).

Improving model robustness and transferability through novel training or processing techniques is therefore a promising and impactful avenue in bioacoustic research. Current approaches include denoising techniques (Denton et al., 2022; Juodakis & Marsland, 2022; Xie et al., 2021), transfer learning with robust models pre-trained on large datasets (Ghani et al., 2023; Hamer et al., 2023), and data aug-

mentation to enhance model robustness against background variability (MacIsaac et al., 2024; Park et al., 2019). Still, significant performance gaps persist in real-world scenarios (Boudiaf et al., 2023; Kahl et al., 2021; Goëau et al., 2018). Future research should prioritise a deeper understanding of the efficacy of these techniques in realistic monitoring scenarios, particularly by incorporating a broader representation of taxa and environmental contexts. Another promising avenue for improving model transferability involves integrating contextual information directly into the inference process. For instance, incorporating environmental covariates—such as habitat type, time of day, season, and geographic location—could enable models to dynamically adapt to varying environmental conditions during inference. Initial research has explored the use of geographic priors in this context (Mac Aodha et al., 2019) or day/night and site covariates (Leseberg et al., 2020); however, the full potential of incorporating a broader range of environmental covariates remains largely untapped. Ultimately, the question of how to effectively adapt bioacoustic models to real-world scenarios will be crucial.

While the ultimate goal of bioacoustic monitoring is to inform ecological understanding and conservation action, this thesis focuses exclusively on the development of models for the detection and classification of individual sound events, as exemplified in Chapters 3 and 4. Acoustic detections provide the foundation for deriving essential metrics in ecological research and conservation management (Gibb et al., 2018), including population density estimates (Pérez-Granados & Traba, 2021), occupancy rates (Wood & Peery, 2022), and species distribution models (Desjonquères et al., 2022). However, employing automated methods for generating these detections introduces additional complexities that must be carefully addressed. Each detection is accompanied by a confidence score which does not directly represent the probability of a correct detection (Dussert et al., 2024) and is influenced by a multitude of factors, including the distance (Knight & Bayne, 2018) and bearing of the vocalisation, variable environmental conditions and noise levels (Leseberg et al., 2020), as well as the inherent characteristics of the model itself (Knight et al., 2017). Consequently, selecting an appropriate confidence score threshold for classifying

detections requires careful manual validation (Wood & Kahl, 2024; Knight et al., 2020) while also considering the specific goals of the study, resulting in a threshold that is often not readily transferable across different species or locations (Navine et al., 2024). Practitioners should be aware of these nuances when interpreting model outputs and recognise that the performance metrics reported during model development may not directly translate to performance in the field.

There is a growing interest in integrating the uncertainties associated with AI model outputs into statistical frameworks used for ecological inference. Examples of this include methods for estimating acoustic activity that directly incorporate model uncertainty (Navine et al., 2024) and occupancy models adapted for AI-derived data (Rhinehart et al., 2022). However, given the numerous decisions involved in developing AI models, it remains unclear how these choices impact the results of such statistical inferences and what implications they have for model development. For instance, Pantazis et al. (2024) found that the choice of model architecture had minimal impact on occupancy estimates derived from automated camera trap detections, while also providing insights into the amount of training data required for reliable estimates. Studies that integrate bioacoustic model development with ecological inference are needed to determine the data requirements for robust performance in specific applications and to guide the development of models that are better aligned with ecological research needs. Furthermore, improving and streamlining validation workflows, potentially through annotation tools like `whombat` and data collection platforms like `acoupi`, will remain crucial for ensuring the reliability of these models in ecological applications.

To further refine ecological metrics like abundance estimates, another important consideration is identifying which vocalisations originate from the same individual. This typically involves using spatial information from multi-sensor arrays, particularly through triangulation of sound source locations, to group vocalisations originating from the same point (Rhinehart et al., 2020; Mesaros et al., 2019; Nguyen et al., 2021). This spatially explicit information can be incorporated into spatial capture-recapture (Wang et al., 2024) or random encounter models (Milchram et

al., 2020) to derive more robust density estimates. Deploying multi-sensor arrays can be challenging, but open-source devices (Heath et al., 2024) are facilitating the collection of data required for localisations and will help the adoption of such approaches. Furthermore, incorporating this device, or a similar one, within the `acoupi` framework could eliminate the often complex post-deployment analysis process needed to obtain these localisations. Alternatively, even with single-sensor recordings, chaining detections into sequences belonging to the same individual could improve the accuracy of abundance estimates. For example, the BatDetect2 model generates detections that are rich in information, including precise timing, call duration, frequency ranges, and a feature vector representing automatically learned acoustic features. This detailed information could be leveraged to identify patterns and consistencies in the acoustic characteristics, timing, and frequency of calls, allowing for the grouping of detections into sequences likely produced by the same individual. However, the development and refinement of these post-processing methodologies are currently limited by the availability of validation data. Incorporating the ability to annotate sequences of sound events in tools such as `whombat` could help address this limitation.

6.4 Conclusions

Acoustic biodiversity monitoring, enhanced by the application of AI, offers the potential to yield rich and detailed insights into species distribution and behaviour, while also enabling monitoring at previously unattainable scales and resolutions. To realise this potential, it is crucial to make AI technology accessible and effective for a broader community of researchers and practitioners. This thesis contributes to this effort by developing and providing user-friendly, open-source tools, including `whombat`, `batdetect2` and `acoupi`, that support the development, validation, and deployment of AI models for bioacoustic monitoring. Fundamental to this process is an understanding of the quantity, quality, and annotation requirements for data, which guides the creation of high-quality datasets and informs the development of effective detection and classification models. Moreover, employing models that

learn more effectively from data can lessen the need for extensive training datasets, and by providing more interpretable results, enhance trust in the model's outputs. Ultimately, the aim of this research is to advance the development and application of AI tools within the toolkit of ecologists and conservationists, thereby supporting biodiversity monitoring across a wide range of contexts.

References

Acoustics, W. (2019). *Kaleidoscope Pro Analysis Software*.

Adelantado, F., X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui & T. Watteyne (2017). "Understanding the Limits of LoRaWAN". In: *IEEE Communications Magazine* 55.9, pages 34–40. ISSN: 1558-1896. DOI: 10.1109/MCOM.2017.1600613.

Aide, T. M., C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega & R. Alvarez (2013). "Real-Time Bioacoustics Monitoring and Automated Species Identification". In: *Peerj* 1, e103. ISSN: 2167-8359. DOI: 10.7717/peerj.103.

Akhtar, M., O. Benjelloun, C. Conforti, P. Gijsbers, J. Giner-Miguelez, N. Jain, M. Kuchnik, Q. Lhoest, P. Marcenac, M. Maskey, P. Mattson, L. Oala, P. Ruyssen, R. Shinde, E. Simperl, G. Thomas, S. Tykhonov, J. Vanschoren, J. van der Velde, S. Vogler & C.-J. Wu (2024). "Croissant: A Metadata Format for ML-Ready Datasets". In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. New York, NY, USA: Association for Computing Machinery, pages 1–6. ISBN: 979-8-4007-0611-0. DOI: 10.1145/3650203.3663326.

Alcocer, I., H. Lima, L. S. M. Sugai & D. Llusia (2022). "Acoustic Indices as Proxies for Biodiversity: A Meta-analysis". In: *Biological Reviews* 97.6, pages 2209–2236. ISSN: 1469-185X. DOI: 10.1111/brv.12890.

Alipek, S., M. Maelzer, Y. Paumen, H. Schauer-Weisshahn & J. Moll (2023). "An Efficient Neural Network Design Incorporating Autoencoders for the Clas-

sification of Bat Echolocation Sounds". In: *Animals* 13.16, page 2560. DOI: 10.3390/ani13162560. pmid: 37627350.

Allen, A. N., M. Harvey, L. Harrell, A. Jansen, K. P. Merkens, C. C. Wall, J. Cattiau & E. M. Oleson (2021). "A Convolutional Neural Network for Automated Detection of Humpback Whale Song in a Diverse, Long-Term Passive Acoustic Dataset". In: *Frontiers in Marine Science* 8. ISSN: 2296-7745. DOI: 10.3389/fmars.2021.607321.

Amyar, A., R. Modzelewski, H. Li & S. Ruan (2020). "Multi-Task Deep Learning Based CT Imaging Analysis for COVID-19 Pneumonia: Classification and Segmentation". In: *Computers in Biology and Medicine* 126, page 104037. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2020.104037.

Audacity, T. (2017). "Audacity". In.

Baker, E. & S. Vincent (2019). "A Deafening Silence: A Lack of Data and Reproducibility in Published Bioacoustics Research?" In: *Biodiversity Data Journal* 7, e36783. ISSN: 1314-2828. DOI: 10.3897/BDJ.7.e36783. pmid: 31723333.

Balle, M., W. Xu, K. F. Darras & T. C. Wanger (2024). "A Power Management and Control System for Environmental Monitoring Devices". In: *IEEE Transactions on AgriFood Electronics*, pages 1–10. ISSN: 2771-9529. DOI: 10.1109/TAFE.2024.3472493.

Banner, K. M., K. M. Irvine, T. J. Rodhouse, W. J. Wright, R. M. Rodriguez & A. R. Litt (2018). "Improving Geographically Extensive Acoustic Survey Designs for Modeling Species Occurrence with Imperfect Detection and Misidentification". In: *Ecology and Evolution* 8.12, pages 6144–6156. ISSN: 2045-7758. DOI: 10.1002/ece3.4162.

Barlow, K. E., P. A. Briggs, K. A. Haysom, A.M. Hutson, A. Hutson, A. M. Hutson, N. Lechiara, P. A. Racey, A. Walsh, S. D. Langton & S. Langton (2015). "Citizen Science Reveals Trends in Bat Populations: The National Bat Monitoring Programme in Great Britain". In: *Biological Conservation* 182, pages 14–26. DOI: 10.1016/j.biocon.2014.11.022.

Bas, Y., D. Bas & J.-F. Julien (2017). "Tadarida: A Toolbox for Animal Detection on Acoustic Recordings". In: *Journal of Open Research Software* 5.1 (1), page 6. ISSN: 2049-9647. DOI: 10.5334/jors.154.

Baucas, M. & P. Spachos (2024). "Edge-Based Data Sensing and Processing Platform for Urban Noise Classification". In: *IEEE Sensors Letters* 8.5, pages 1–4. DOI: 10.1109/LSENS.2024.3392163.

Baucas, M. J. & P. Spachos (2020). "Using Cloud and Fog Computing for Large Scale IoT-based Urban Sound Classification". In: *Simulation Modelling Practice and Theory*. Modeling and Simulation of Fog Computing 101, page 102013. ISSN: 1569-190X. DOI: 10.1016/j.simpat.2019.102013.

Beery, S., D. Morris, S. Yang, M. Simon, A. Norouzzadeh & N. Joshi (2019). "Efficient Pipeline for Automating Species ID in New Camera Trap Projects". In: *Biodiversity Information Science and Standards* 3, e37222. ISSN: 2535-0897. DOI: 10.3897/biss.3.37222.

Bermant, P. C., M. M. Bronstein, R. J. Wood, S. Gero & D. F. Gruber (2019). "Deep Machine Learning Techniques for the Detection and Classification of Sperm Whale Bioacoustics". In: *Scientific Reports* 9.1, page 12588. ISSN: 2045-2322. DOI: 10.1038/s41598-019-48909-4.

Besson, M., J. Alison, K. Bjerge, T. E. Gorochowski, T. T. Høye, T. Jucker, H. M. R. Mann & C. F. Clements (2022). "Towards the Fully Automated Monitoring of Ecological Communities". In: *Ecology Letters* 25.12, pages 2753–2775. ISSN: 1461-0248. DOI: 10.1111/ele.14123.

Bick, I. A., V. Bakkestuen, B. Cretois, B. Hillier, J. A. Kålås, M. Pedersen, K. Raja, C. M. Rosten, M. Somveille, B. G. Stokke, J. Wiel & S. S. Sethi (2024). *National-Scale Acoustic Monitoring of Avian Biodiversity and Migration*. DOI: 10.1101/2024.05.21.595242. URL: http://biorxiv.org/lookup/doi/10.1101/2024.05.21.595242 (visited on 05/12/2024). Pre-published.

Bidarouni, A. L. & J. Abeßer (2024). "Towards Domain Shift in Location-Mismatch Scenarios for Bird Activity Detection". In: *2024 32nd European Signal Processing Conference (EUSIPCO)*. 2024 32nd European Signal Processing Con-

ference (EUSIPCO), pages 1267–1271. DOI: 10.23919/EUSIPCO63174.2024.10715313.

Bierman, G., M. Abadi & M. Torgersen (2014). "Understanding TypeScript". In: *ECOOP 2014 – Object-Oriented Programming*. Edited by R. Jones. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pages 257–281. ISBN: 978-3-662-44202-9. DOI: 10.1007/978-3-662-44202-9_11.

Borowiec, M. L., R. B. Dikow, P. B. Frandsen, A. McKeeken, G. Valentini & A. E. White (2022). "Deep Learning as a Tool for Ecology and Evolution". In: *Methods in Ecology and Evolution* 13.8, pages 1640–1660. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13901.

Boudiaf, M., T. Denton, B. van Merriënboer, V. Dumoulin & E. Triantafillou (2023). *In Search for a Generalizable Method for Source Free Domain Adaptation*. DOI: 10.48550/arXiv.2302.06658. arXiv: 2302.06658 [cs]. URL: http://arxiv.org/abs/2302.06658 (visited on 22/02/2024). Pre-published.

Bradfer-Lawrence, T., B. Duthie, C. Abrahams, M. Adam, R. J. Barnett, A. Beeston, J. Darby, B. Dell, N. Gardner, A. Gasc, B. Heath, N. Howells, M. Janson, M.-V. Kyoseva, T. Luypaert, O. C. Metcalf, A. E. Nousek-McGregor, F. Poznansky, S. R. P.-J. Ross, S. Sethi, S. Smyth, E. Waddell & J. S. P. Froidevaux (2024). "The Acoustic Index User's Guide: A Practical Manual for Defining, Generating and Understanding Current and Future Acoustic Indices". In: *Methods in Ecology and Evolution* n/a.n/a (). ISSN: 2041-210X. DOI: 10.1111/2041-210X.14357.

Bradfer-Lawrence, T., N. Gardner, L. Bunnefeld, N. Bunnefeld, S. G. Willis & D. H. Dent (2019). "Guidelines for the Use of Acoustic Indices in Environmental Research". In: *Methods in Ecology and Evolution* 10.10, pages 1796–1807. ISSN: 2041-210X. DOI: 10.1111/2041-210x.13254.

Bravo Sanchez, F. J., M. R. Hossain, N. B. English & S. T. Moore (2021). "Bioacoustic Classification of Avian Calls from Raw Sound Waveforms with an Open-Source Deep Learning Architecture". In: *Scientific Reports* 11.1, page 15733. ISSN: 2045-2322. DOI: 10.1038/s41598-021-95076-6.

Breiman, L. (2001). “Random Forests”. In: *Machine Learning* 45.1, pages 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

Brinkløv, S. M. M., J. Macaulay, C. Bergler, J. Tougaard, K. Beedholm, M. Elmeros & P. T. Madsen (2023). “Open-Source Workflow Approaches to Passive Acoustic Monitoring of Bats”. In: *Methods in Ecology and Evolution* 14.7, pages 1747–1763. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14131.

Browning, E., R. Gibb, P. Glover-Kapfer & K. E. Jones (2017). *Passive Acoustic Monitoring in Ecology and Conservation*. WWF-UK, page 75.

Brunoldi, M., G. Bozzini, A. Casale, P. Corvisiero, D. Grosso, N. Magnoli, J. Alessi, C. N. Bianchi, A. Mandich, C. Morri, P. Povero, M. Wurtz, C. Melchiorre, G. Viano, V. Cappanera, G. Fanciulli, M. Bei, N. Stasi & M. Taiuti (2016). “A Permanent Automated Real-Time Passive Acoustic Monitoring System for Bottlenose Dolphin Conservation in the Mediterranean Sea”. In: *PLOS ONE* 11.1, e0145362. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0145362.

Budach, L., M. Feuerpfeil, N. Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann & H. Harmouch (2022). *The Effects of Data Quality on Machine Learning Performance*. DOI: 10.48550/arXiv.2207.14529. arXiv: 2207.14529 [cs]. URL: http://arxiv.org/abs/2207.14529 (visited on 28/11/2024). Pre-published.

Bull, J. W., E. J. Milner-Gulland, P. F. E. Addison, W. N. S. Arlidge, J. Baker, T. M. Brooks, M. J. Burgass, A. Hinsley, M. Maron, J. G. Robinson, N. Sekhran, S. P. Sinclair, S. N. Stuart, S. O. S. E. zu Ermgassen & J. E. M. Watson (2019). “Net Positive Outcomes for Nature”. In: *Nature Ecology & Evolution* 4.1, pages 4–7. DOI: 10.1038/s41559-019-1022-z. pmid: 31686021.

Cañas, J. S., M. P. Toro-Gómez, L. S. M. Sugai, H. D. Benítez Restrepo, J. Rudas, B. Posso Bautista, L. F. Toledo, S. Dena, A. H. R. Domingos, F. L. de Souza, S. Neckel-Oliveira, A. da Rosa, V. Carvalho-Rocha, J. V. Bernardy, J. L. M. M. Sugai, C. E. dos Santos, R. P. Bastos, D. Llusia & J. S. Ulloa (2023). “A Dataset for Benchmarking Neotropical Anuran Calls Identification in Passive

Acoustic Monitoring". In: *Scientific Data* 10.1, page 771. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02666-2.

Cannam, C., C. Landone & M. Sandler (2010). "Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files". In: *Proceedings of the ACM Multimedia 2010 International Conference*. Firenze, Italy, pages 1467–1468.

Cartwright, M., G. Dove, A. E. Méndez Méndez, J. P. Bello & O. Nov (2019). "Crowdsourcing Multi-label Audio Annotation Tasks with Citizen Scientists". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19: CHI Conference on Human Factors in Computing Systems. Glasgow Scotland Uk: ACM, pages 1–11. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300522.

Cauchy, P., K. J. Heywood, D. Risch, N. D. Merchant, B. Y. Queste & P. Testor (2020). "Sperm Whale Presence Observed Using Passive Acoustic Monitoring from Gliders of Opportunity". In: *Endangered Species Research* 42, pages 133–149. DOI: 10.3354/esr01044.

CBD, U. N. (2022). "Kunming-Montreal Global Biodiversity Framework". In: *Fifteenth Meeting of the Conference of the Parties to the Convention on Biological Diversity (Part Two) Decision 15/4*. UN Environment Programme Montreal, Canada.

Chasmai, M., A. Shepard, S. Maji & G. V. Horn (2024). "The iNaturalist Sounds Dataset". In: The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Chen, S.-H., J.-C. Wang, H.-J. Lin, M.-H. Lee, A.-C. Liu, Y.-L. Wu, P.-S. Hsu, E.-C. Yang & J.-A. Jiang (2024). "A Machine Learning-Based Multiclass Classification Model for Bee Colony Anomaly Identification Using an IoT-based Audio Monitoring System with an Edge Computing Framework". In: *Expert Systems with Applications* 255, page 124898. DOI: 10.1016/j.eswa.2024.124898.

Chen, X., J. Zhao, Y.-h. Chen, W. Zhou & A. C. Hughes (2020). "Automatic Standardized Processing and Identification of Tropical Bat Calls Using Deep Learning Approaches". In: *Biological Conservation* 241, page 108269. ISSN: 0006-3207. DOI: 10.1016/j.biocon.2019.108269.

Cheng, H., M. Zhang & J. Q. Shi (2024). "A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12, pages 10558–10578. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2024.3447085.

Christin, S., É. Hervet & N. Lecomte (2019). "Applications for Deep Learning in Ecology". In: *Methods in Ecology and Evolution* 10.10, pages 1632–1644. DOI: 10.1111/2041-210x.13256.

Christin, S., É. Hervet, P. Smith, R. Alisauskas, D. Berteaux, G. Brown, K. Elliott, J. Hansen, S. Lai, J.-F. Lamarre, R. Lanctot, C. Latty, A. L. Pogam, D. MacNearney, V. Patil, J. Rausch, S. Saalfeld, N. Schmidt, A. Tam, F. Vézina, Ø. Varpe, P. Woodard, G. Yannic & N. Lecomte (2023). *Deep Learning for Passive Acoustic Monitoring: How to Study Changing Phenology in Remote Areas*. DOI: 10.22541/au.169963215.50290219/v1. URL: https://www.authorea.com/users/369228/articles/686097-deep-learning-for-passive-acoustic-monitoring-how-to-study-changing-phenology-in-remote-areas?commit=c82360f0a2424833c0cf054601cf987171e49e84 (visited on 22/02/2024). Pre-published.

Clark, T., V. Parashchak & S. Pohlenz (2023). *BirdWeather*. BirdWeather. URL: https://app.birdweather.com/ (visited on 05/12/2024).

Coffey, K. R., R. E. Marx & J. F. Neumaier (2019). "DeepSqueak: A Deep Learning-Based System for Detection and Analysis of Ultrasonic Vocalizations". In: *Neuropsychopharmacology* 44.5, pages 859–868. ISSN: 1740-634X. DOI: 10.1038/s41386-018-0303-6. pmid: 30610191.

Cole, E., X. Yang, K. Wilber, O. Mac Aodha & S. Belongie (2022). "When Does Contrastive Visual Representation Learning Work?" In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14755–14764.

Collen, B., M. Ram, T. Zamin & L. McRae (2008). "The Tropical Biodiversity Data Gap: Addressing Disparity in Global Monitoring". In: *Tropical Conservation Science* 1.2, pages 75–88. ISSN: 1940-0829. DOI: 10.1177/194008290800100202.

Conservation Bioacoustics, K. L. Y. C. for (2023). *Raven Pro*. Version 1.6.5. K. Lisa Yang Center for Conservation Bioacoustics.

Dalziell, A. H., J. A. Welbergen, B. Igic & R. D. Magrath (2014). "Avian Vocal Mimicry: A Unified Conceptual Framework". In: *Biological Reviews* 90.2, pages 643–668. ISSN: 1469-185X. DOI: 10.1111/brv.12129.

Darras, K., B. Kolbrek, A. Knorr, V. Meyer, M. Zippert & A. Wenzel (2021). "Assembling Cheap, High-Performance Microphones for Recording Terrestrial Wildlife: The Sonitor System". In: *F1000Research* 7, page 1984. ISSN: 2046-1402. DOI: 10.12688/f1000research.17511.3. pmid: 30687500.

Darras, K. F. A., M. Balle, W. Xu, Y. Yan, V. G. Zakka, M. Toledo-Hernández, D. Sheng, W. Lin, B. Zhang, Z. Lan, L. Fupeng & T. C. Wanger (2024). "Eyes on Nature: Embedded Vision Cameras for Terrestrial Biodiversity Monitoring". In: *Methods in Ecology and Evolution* 15.12, pages 2262–2275. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14436.

Denton, T., S. Wisdom & J. R. Hershey (2022). "Improving Bird Classification with Unsupervised Sound Separation". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 636–640. DOI: 10.1109/ICASSP43922.2022.9747202.

Denzinger, A. & H.-U. Schnitzler (2013). "Bat Guilds, a Concept to Classify the Highly Diverse Foraging and Echolocation Behaviors of Microchiropteran Bats". In: *Frontiers in Physiology* 4. ISSN: 1664-042X. DOI: 10.3389/fphys.2013.00164.

Desislavov, R., F. Martínez-Plumed & J. Hernández-Orallo (2023). "Compute and Energy Consumption Trends in Deep Learning Inference". In: *Sustainable Computing: Informatics and Systems* 38, page 100857. ISSN: 22105379. DOI: 10.1016/j.suscom.2023.100857. arXiv: 2109.05472 [cs].

Desjonquères, C., S. Linke, J. Greenhalgh, F. Rybak & J. Sueur (2024). "The Potential of Acoustic Monitoring of Aquatic Insects for Freshwater Assessment". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 379.1904. DOI: 10.1098/rstb.2023.0109. pmid: 38705188.

Desjonquères, C., S. Villén-Pérez, P. De Marco, R. Márquez, J. F. Beltrán & D. Llusia (2022). "Acoustic Species Distribution Models (aSDMs): A Framework to Forecast Shifts in Calling Behaviour under Climate Change". In: *Methods in Ecology and Evolution* 13.10, pages 2275–2288. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13923.

Díaz, S., J. Settele, E. S. Brondízio, H. T. Ngo, J. Agard, A. Arneth, P. Balvanera, K. A. Brauman, S. H. M. Butchart, K. M. A. Chan, L. A. Garibaldi, K. Ichii, J. Liu, S. M. Subramanian, G. F. Midgley, P. Miloslavich, Z. Molnár, D. Obura, A. Pfaff, S. Polasky, A. Purvis, J. Razzaque, B. Reyers, R. R. Chowdhury, Y.-J. Shin, I. Visseren-Hamakers, K. J. Willis & C. N. Zayas (2019). "Pervasive Human-Driven Decline of Life on Earth Points to the Need for Transformative Change". In: *Science* 366.6471, page 3100. DOI: 10.1126/science.aax3100. pmid: 31831642.

Dierckx, L., M. Beauvois & S. Nijssen (2022). "Detection and Multi-label Classification of Bats". In: *Advances in Intelligent Data Analysis XX*. Edited by T. Bouadi, E. Fromont & E. Hüllermeier. Lecture Notes in Computer Science. Cham: Springer International Publishing, pages 53–65. ISBN: 978-3-031-01333-1. DOI: 10.1007/978-3-031-01333-1_5.

Disabato, S., G. Canonaco, P. G. Flikkema, M. Roveri & C. Alippi (2021). "Birdsong Detection at the Edge with Deep Learning". In: *International Conference on Smart Computing*. DOI: 10.1109/smartcomp52413.2021.00022.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit & N. Houlsby (2021). "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv:2010.11929 [cs.CV]*. DOI: 10.48550/arXiv.2010.11929. arXiv: 2010.11929 [cs.CV].

Dufourq, E., C. Batist, R. Foquet & I. Durbach (2022). "Passive Acoustic Monitoring of Animal Populations with Transfer Learning". In: *Ecological Informatics* 70, page 101688. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2022.101688. pmid: null.

Dufourq, E., I. Durbach, J. P. Hansford, A. Hoepfner, H. Ma, J. V. Bryant, C. S. Stender, W. Li, Z. Liu, Q. Chen, Z. Zhou & S. T. Turvey (2021). "Automated Detection of Hainan Gibbon Calls for Passive Acoustic Monitoring". In: *Remote Sensing in Ecology and Conservation* 7.3, pages 475–487. ISSN: 2056-3485. DOI: 10.1002/rse2.201.

Dussert, G., S. Chamaillé-Jammes, S. Dray & V. Miele (2024). "Being Confident in Confidence Scores: Calibration in Deep Learning Models for Camera Trap Image Sequences". In: *Remote Sensing in Ecology and Conservation* n/a.n/a. ISSN: 2056-3485. DOI: 10.1002/rse2.412.

Espy, M. & B. Babbitt (1994). "Record of Decision for Amendments to Forest Service and Bureau of Land Management Planning Documents within the Range of the Northern Spotted Owl". In: *US Department of Agriculture, Forest Service and US Department of the Interior, Bureau of Land Management, Washington, DC*.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn & A. Zisserman (2009). "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2, pages 303–338. DOI: 10.1007/s11263-009-0275-4.

Fairbrass, A. J., M. Firman, C. Williams, G. J. Brostow, H. Titheridge & K. E. Jones (2019). "CityNet—Deep Learning Tools for Urban Ecoacoustic Assessment". In: *Methods in Ecology and Evolution* 10.2, pages 186–197. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13114.

Fairbrass, A. J., P. Rennert, C. Williams, H. Titheridge & K. E. Jones (2017). "Biases of Acoustic Indices Measuring Biodiversity in Urban Areas". In: *Ecological Indicators* 83, pages 169–177. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2017.07.064.

Falcon, W. & The PyTorch Lightning team (2019). *PyTorch Lightning*. Version 1.4. DOI: 10.5281/zenodo.3828935.

Farley, S. S., A. Dawson, S. J. Goring & J. W. Williams (2018). "Situating Ecology as a Big-Data Science: Current Advances, Challenges, and Solutions". In: *BioScience* 68.8, pages 563–576. DOI: 10.1093/biosci/biy068.

Farmer, R. G., M. L. Leonard & A. G. Horn (2012). "Observer Effects and Avian-Call-Count Survey Quality: Rare-Species Biases and Overconfidence". In: *The Auk* 129.1, pages 76–86. DOI: 10.1525/auk.2012.11129.

Ferreira, D. F., R. Gibb, A. López-Baucells, N. J. Nunes, K. E. Jones & R. Rocha (2022). "Species-Specific Responses to Land-Use Change in Island Insectivorous Bats". In: *Journal for Nature Conservation* 67, page 126177. DOI: 10.1016/j.jnc.2022.126177.

Ford, H. V., F. Schrodt, A. Zieritz, D. A. Exton, G. van der Heijden, J. Teague, T. Coles & R. Field (2024). "A Technological Biodiversity Monitoring Toolkit for Biocredits". In: *Journal of Applied Ecology* 61.9, pages 2007–2019. ISSN: 1365-2664. DOI: 10.1111/1365-2664.14725.

Franklin, A. B., K. M. Dugger, D. B. Lesmeister, R. J. Davis, J. D. Wiens, G. C. White, J. D. Nichols, J. E. Hines, C. B. Yackulic, C. J. Schwarz, S. H. Ackers, L. S. Andrews, L. L. Bailey, R. Bown, J. Burgher, K. P. Burnham, P. C. Carlson, Tara Chestnut, Tara Chestnut, T. Chestnut, M. M. Conner, K. E. Dilione, E. D. Forsman, E. D. Forsman, E. M. Glenn, S. A. Gremel, Scott A. Gremel, K. E. Hamm, K. A. Hamm, D. R. Herter, J. M. Higley, Rob B. Horn, R. B. Horn, J. M. A. Jenkins, W. L. Kendall, W. L. Kendall, D. W. Lamphear, C. McCafferty, Christopher McCafferty, T. L. McDonald, J. A. Reid, J. T. Rockweit, D. C. Simon, S. G. Sovern, Stan G. Sovern, James K. Swingle, James K. Swingle, J. K. Swingle & H. Wise (2021). "Range-Wide Declines of Northern Spotted

Owl Populations in the Pacific Northwest: A Meta-Analysis". In: *Biological Conservation* 259, page 109168. DOI: 10.1016/j.biocon.2021.109168.

Fraser, E. E. (2018). "Manual Analysis of Recorded Bat Echolocation Calls: Summary, Synthesis, and Proposal for Increased Standardization in Training Practices". In: *Canadian Journal of Zoology* 96.6, pages 505–512. ISSN: 0008-4301. DOI: 10.1139/cjz-2017-0175.

Frick, W. F., T. Kingston & J. Flanders (2019). "A Review of the Major Threats and Challenges to Global Bat Conservation". In: *Annals of the New York Academy of Sciences* 1469.1, pages 5–25. ISSN: 1749-6632. DOI: 10.1111/nyas.14045.

Fundel, F., D. A. Braun & S. Gottwald (2023). "Automatic Bat Call Classification Using Transformer Networks". In: *Ecological Informatics* 78, page 102288. DOI: 10.1016/j.ecoinf.2023.102288.

Gallacher, S., D. Wilson, A. Fairbrass, D. Turmukhambetov, M. Firman, S. Kreitmayer, O. Mac Aodha, G. Brostow & K. Jones (2021). "Shazam for Bats: Internet of Things for Continuous Real-time Biodiversity Monitoring". In: *IET Smart Cities* 3.3, pages 171–183. DOI: 10.1049/smc2.12016.

Ganchev, T., I. Potamitis & N. Fakotakis (2007). "Acoustic Monitoring of Singing Insects". In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07* 4, pages IV-721-IV–724. DOI: 10.1109/icassp.2007.367014.

Gasc, A., J. Sueur, F. Jiguet, V. Devictor, P. Grandcolas, C. Burrow, M. Depraetere & S. Pavoine (2013). "Assessing Biodiversity with Sound: Do Acoustic Diversity Indices Reflect Phylogenetic and Functional Diversities of Bird Communities?" In: *Ecological Indicators* 25, pages 279–287. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2012.10.009.

Gemmeke, J. F., D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal & M. Ritter (2017). "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). New Orleans,

LA: IEEE, pages 776–780. ISBN: 978-1-5090-4117-6. DOI: 10.1109/ICASSP.2017.7952261.

Ghani, B., T. Denton, S. Kahl & H. Klinck (2023). "Global Birdsong Embeddings Enable Superior Transfer Learning for Bioacoustic Classification". In: *Scientific Reports* 13.1. DOI: 10.1038/s41598-023-49989-z. pmid: 38129622.

Gibb, R., E. Browning, P. Glover-Kapfer & K. E. Jones (2018). "Emerging Opportunities and Challenges for Passive Acoustics in Ecological Assessment and Monitoring". In: *Methods in Ecology and Evolution* 10.2 (2), pages 169–185. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13101.

Goëau, H., S. Kahl, H. Glotin, R. Planqué, W.-P. Vellinga & A. Joly (2018). "Overview of BirdCLEF 2018: Monospecies vs. Soundscape Bird Identification". In: *CEUR Workshops Proceedings*. Volume 2125. CEUR Workshops Proceedings 9. Avignon, France.

Gomes, D. G. E., P. Pottier, R. Crystal-Ornelas, E. J. Hudgins, V. Foroughirad, L. L. Sánchez-Reyes, R. Turba, P. A. Martinez, D. Moreau, M. G. Bertram, C. A. Smout & K. M. Gaynor (2022). "Why Don't We Share Data and Code? Perceived Barriers and Benefits to Public Archiving Practices". In: *Proceedings of the Royal Society B: Biological Sciences* 289.1987, page 20221113. DOI: 10.1098/rspb.2022.1113.

Gonzalez, A., J. M. Chase & M. I. O'Connor (2023). "A Framework for the Detection and Attribution of Biodiversity Change". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 378.1881. DOI: 10.1098/rstb.2022.0182. pmid: 37246383.

Görföl, T., J. C.-C. Huang, G. Csorba, D. Győrössy, P. Estók, T. Kingston, K. L. Szabadi, E. McArthur, J. Senawi, N. M. Furey, V. T. Tu, V. D. Thong, F. A. A. Khan, E. R. Jinggong, M. Donnelly, J. V. Kumaran, J.-N. Liu, S.-F. Chen, M.-N. Tuanmu, Y.-Y. Ho, H.-C. Chang, N.-A. Elias, N.-I. Abdullah, L.-S. Lim, C. D. Squire & S. Zsebők (2022). "ChiroVox: A Public Library of Bat Calls". In: *PeerJ* 10, e12445. ISSN: 2167-8359. DOI: 10.7717/peerj.12445.

Gou, J., B. Yu, S. J. Maybank & D. Tao (2021). "Knowledge Distillation: A Survey". In: *International Journal of Computer Vision* 129.6, pages 1789–1819. ISSN: 1573-1405. DOI: 10.1007/s11263-021-01453-z.

Guo, C., G. Pleiss, Y. Sun & K. Q. Weinberger (2017). *On Calibration of Modern Neural Networks*. arXiv: 1706.04599 [cs]. URL: http://arxiv.org/abs/1706.04599 (visited on 16/05/2022). Pre-published.

Gupta, G., M. Kshirsagar, M. Zhong, S. Gholami & J. L. Ferres (2021). "Comparing Recurrent Convolutional Neural Networks for Large Scale Bird Species Classification". In: *Scientific Reports* 11.1, page 17085. ISSN: 2045-2322. DOI: 10.1038/s41598-021-96446-w.

Hagiwara, M. (2023). "AVES: Animal Vocalization Encoder Based on Self-Supervision". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. DOI: 10.1109/icassp49357.2023.10095642.

Hagiwara, M., B. Hoffman, J.-Y. Liu, M. Cusimano, F. Effenberger & K. Zacarian (2023). "BEANS: The Benchmark of Animal Sounds". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. DOI: 10.1109/ICASSP49357.2023.10096686.

Hall, M. & D. Robinson (2021). "Chapter One - Acoustic Signalling in Orthoptera". In: *Advances in Insect Physiology*. Edited by R. Jurenka. Volume 61. Sound Communication in Insects. Academic Press, pages 1–99. DOI: 10.1016/bs.aiip.2021.09.001.

Hamer, J., E. Triantafillou, B. van Merriënboer, S. Kahl, H. Klinck, T. Denton & V. Dumoulin (2023). "BIRB: A Generalization Benchmark for Information Retrieval in Bioacoustics". In: *arXiv.org*. DOI: 10.48550/arxiv.2312.07439.

Hampton, S. E., S. S. Anderson, S. C. Bagby, C. Gries, X. Han, E. M. Hart, M. B. Jones, W. C. Lenhardt, A. MacDonald, W. K. Michener, J. Mudge, A. Pourmokhtarian, M. P. Schildhauer, K. H. Woo & N. Zimmerman (2015). "The Tao

of Open Science for Ecology". In: *Ecosphere* 6.7, art120. ISSN: 2150-8925. DOI: 10.1890/ES14-00402.1.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke & T. E. Oliphant (2020). "Array Programming with NumPy". In: *Nature* 585.7825 (7825), pages 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2.

He, K., X. Zhang, S. Ren & J. Sun (2016). "Deep Residual Learning for Image Recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778.

Heath, B. E., R. Suzuki, N. P. Le Penru, J. Skinner, C. D. L. Orme, R. M. Ewers, S. S. Sethi & L. Picinali (2024). "Spatial Ecosystem Monitoring with a Multichannel Acoustic Autonomous Recording Unit (MAARU)". In: *Methods in Ecology and Evolution* 15.9, pages 1568–1579. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14390.

Heggan, C., S. Budgett, T. Hospedales & M. Yaghoobi (2024). *On the Transferability of Large-Scale Self-Supervision to Few-Shot Audio Classification*. DOI: 10.48550/arXiv.2402.01274. arXiv: 2402.01274 [cs]. URL: http://arxiv.org/abs/2402.01274 (visited on 04/12/2024). Pre-published.

Hershey, S., S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss & K. Wilson (2017). "CNN Architectures for Large-Scale Audio Classification". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 131–135. DOI: 10.1109/ICASSP.2017.7952132.

Hershey, S., D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. Channing Moore & M. Plakal (2021). "The Benefit of Temporally-Strong Labels in Audio Event

Classification". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 366–370. DOI: 10.1109/ICASSP39728.2021.9414579.

Hill, A. P., P. Prince, E. Piña Covarrubias, C. P. Doncaster, J. L. Snaddon & A. Rogers (2018). "AudioMoth: Evaluation of a Smart Open Acoustic Device for Monitoring Biodiversity and the Environment". In: *Methods in Ecology and Evolution* 9.5, pages 1199–1211. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12955.

Hill, A. P., P. Prince, J. L. Snaddon, C. P. Doncaster & A. Rogers (2019). "AudioMoth: A Low-Cost Acoustic Device for Monitoring Biodiversity and the Environment". In: *HardwareX* 6, e00073. ISSN: 2468-0672. DOI: 10.1016/j.ohx.2019.e00073.

Höchst, J., H. Bellafkir, P. Lampe, M. Vogelbacher, M. Mühling, D. Schneider, K. Lindner, S. Rösner, D. G. Schabo, N. Farwig & B. Freisleben (2022). "Bird@Edge: Bird Species Recognition at the Edge". In: *Networked Systems*. Edited by M.-A. Koulali & M. Mezini. Cham: Springer International Publishing, pages 69–86. ISBN: 978-3-031-17436-0. DOI: 10.1007/978-3-031-17436-0_6.

Hoefer, S., D. T. McKnight, S. Allen-Ankins, E. J. Nordberg & L. Schwarzkopf (2023). "Passive Acoustic Monitoring in Terrestrial Vertebrates: A Review". In: *Bioacoustics* 32.5, pages 506–531. DOI: 10.1080/09524622.2023.2209052.

Hoggatt, M. L., C. A. Starbuck & J. M. O'Keefe (2024). "Acoustic Monitoring Yields Informative Bat Population Density Estimates". In: *Ecology and Evolution* 14.2, e11051. ISSN: 2045-7758. DOI: 10.1002/ece3.11051. pmid: 38389998.

Hohman, F., K. Wongsuphasawat, M. B. Kery & K. Patel (2020). "Understanding and Visualizing Data Iteration in Machine Learning". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13. DOI: 10.1145/3313831.3376177.

Horton, K. G., C. Nilsson, B. M. Van Doren, F. A. La Sorte, A. M. Dokter & A. Farnsworth (2019). "Bright Lights in the Big Cities: Migratory Birds' Exposure to Artificial Light". In: *Frontiers in Ecology and the Environment* 17.4, pages 209–214. ISSN: 1540-9309. DOI: 10.1002/fee.2029.

Hua, H., Y. Li, T. Wang, N. Dong, W. Li & J. Cao (2023). "Edge Computing with Artificial Intelligence: A Machine Learning Perspective". In: *ACM Comput. Surv.* 55.9, 184:1–184:35. ISSN: 0360-0300. DOI: 10.1145/3555802.

INEGI CONABIO, I. N. E. (2008). "Ecorregiones Terrestres de México". In: *Escala* 1.

Ioffe, S. & C. Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". arXiv: 1502.03167 [cs].

IPBES (2019). *Global Assessment Report on Biodiversity and Ecosystem Services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services*. Zenodo. DOI: 10.5281/zenodo.5657041.

Jakobsen, L., S. Brinkløv & A. Surlykke (2013). "Intensity and Directionality of Bat Echolocation Signals". In: *Frontiers in Physiology* 4. ISSN: 1664-042X. DOI: 10.3389/fphys.2013.00089.

Jarrahi, M. H., A. Memariani & S. Guha (2022). "The Principles of Data-Centric AI (DCAI)". In: *arXiv*. DOI: 10.48550/arxiv.2211.14611.

Jolles, J. W. (2021). "Broad-scale Applications of the Raspberry Pi: A Review and Guide for Biologists". In: *Methods in Ecology and Evolution* 12.9, pages 1562–1579. DOI: 10.1111/2041-210x.13652.

Jones, G. & M. W. Holderied (2007). "Bat Echolocation Calls: Adaptation and Convergent Evolution". In: *Proceedings of the Royal Society B: Biological Sciences* 274.1612, pages 905–912. DOI: 10.1098/rspb.2006.0200.

Jones, G., D. S. Jacobs, T. H. Kunz, M. R. Willig & P. A. Racey (2009). "Carpe Noctem: The Importance of Bats as Bioindicators". In: *Endangered Species Research* 8.1–2, pages 93–115. ISSN: 1863-5407, 1613-4796. DOI: 10.3354/esr00182.

Jones, G. & B. M. Siemers (2011). "The Communicative Potential of Bat Echolocation Pulses". In: *Journal of Comparative Physiology A* 197.5, pages 447–457. ISSN: 1432-1351. DOI: 10.1007/s00359-010-0565-x.

Jones, G. & E. Teeling (2006). "The Evolution of Echolocation in Bats". In: *Trends in Ecology & Evolution* 21.3, pages 149–156. ISSN: 0169-5347. DOI: 10.1016/j.tree.2006.01.001.

Jones, K. E., J. A. Russ, A.-T. Bashta, Z. Bilhari, C. Catto, I. Csősz, A. Gorbachev, P. Győrfi, A. Hughes, I. Ivashkiv, N. Koryagina, A. Kurali, S. Langton, A. Collen, G. Margiean, I. Pandourski, S. Parsons, I. Prokofev, A. Szodoray-Paradi, F. Szodoray-Paradi, E. Tilova, C. L. Walters, A. Weatherill & O. Zavarzin (2013). "Indicator Bats Program: A System for the Global Acoustic Monitoring of Bats". In: *Biodiversity Monitoring and Conservation*. John Wiley & Sons, Ltd, pages 211–247. ISBN: 978-1-118-49074-7. DOI: 10.1002/9781118490747.ch10.

Juodakis, J. & S. Marsland (2022). "Wind-Robust Sound Event Detection and Denoising for Bioacoustics". In: *Methods in Ecology and Evolution* 13.9, pages 2005–2017. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13928.

Kahl, S., C. M. Wood, M. Eibl & H. Klinck (2021). "BirdNET: A Deep Learning Solution for Avian Diversity Monitoring". In: *Ecological Informatics* 61, page 101236. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2021.101236.

Kalan, A. K., R. Mundry, O. J. Wagner, S. Heinicke, C. Boesch & H. S. Kühl (2015). "Towards the Automated Detection and Occupancy Estimation of Primates Using Passive Acoustic Monitoring". In: *Ecological Indicators* 54, pages 217–226. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2015.02.023.

Kamminga, J., E. Ayele, N. Meratnia & P. Havinga (2018). "Poaching Detection Technologies—A Survey". In: *Sensors* 18.5 (5), page 1474. ISSN: 1424-8220. DOI: 10.3390/s18051474.

Kandel, S., A. Paepcke, J. M. Hellerstein & J. Heer (2012). "Enterprise Data Analysis and Visualization: An Interview Study". In: *IEEE Transactions on Visualization*

and Computer Graphics 18.12, pages 2917–2926. DOI: 10.1109/tvcg.2012.219. pmid: 26357201.

Kaplan, J., S. McCandlish, Tom Henighan, T. Henighan, T. B. Brown, B. Chess, R. Child, Rewon Child, S. Gray, A. Radford, J. Wu, D. Amodei & Dario Amodei (2020). "Scaling Laws for Neural Language Models". In: *arXiv: Learning*.

Karlsson, E. C. M., H. Tay, P. Imbun & A. C. Hughes (2021). "The Kinabalu Recorder, a New Passive Acoustic and Environmental Monitoring Recorder". In: *Methods in Ecology and Evolution* 12.11, pages 2109–2116. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13671.

Kath, H., P. P. Serafini, I. B. Campos, T. S. Gouvêa & D. Sonntag (2024). "Leveraging transfer learning and active learning for data annotation in passive acoustic monitoring of wildlife". In: *Ecological Informatics* 82, page 102710.

Kerbiriou, C., J. F. Julien, Y. Bas, J. Marmet, R. Lorrilliere, C. Azam, A. Gasc & G. Lois (2015). "Vigie-Chiro : 9 ans de suivi des tendances des espèces communes". In: *Symbioses*, pages 34, 35.

Kershenbaum, A., D. T. Blumstein, M. A. Roch, Ç. Akçay, G. Backus, M. A. Bee, K. Bohn, Y. Cao, G. Carter, C. Cäsar, M. Coen, S. L. DeRuiter, L. Doyle, S. Edelman, R. Ferrer-i-Cancho, T. M. Freeberg, E. C. Garland, M. Gustison, H. E. Harley, C. Huetz, M. Hughes, J. Hyland Bruno, A. Ilany, D. Z. Jin, M. Johnson, C. Ju, J. Karnowski, B. Lohr, M. B. Manser, B. McCowan, E. Mercado, P. M. Narins, A. Piel, M. Rice, R. Salmi, K. Sasahara, L. Sayigh, Y. Shiu, C. Taylor, E. E. Vallejo, S. Waller & V. Zamora-Gutierrez (2014). "Acoustic Sequences in Non-human Animals: A Tutorial Review and Prospectus". In: *Biological Reviews* 91.1, pages 13–52. ISSN: 1469-185X. DOI: 10.1111/brv.12160.

Khalighifar, A., B. S. Gotthold, E. Adams, J. Barnett, L. O. Beard, E. R. Britzke, P. A. Burger, K. Chase, Z. Cordes, P. M. Cryan, E. Ferrall, C. T. Fill, S. E. Gibson, G. S. Haulton, K. M. Irvine, L. S. Katz, W. L. Kendall, C. A. Long, O. Mac Aodha, T. McBurney, S. McCarthy, M. W. McKown, J. O'Keefe, L. D. Patterson, K. A. Pitcher, M. Rustand, J. L. Segers, K. Seppanen, J. L. Siemers, C. Stratton, B. R. Straw, T. J. Weller & B. E. Reichert (2022). "NABat ML:

Utilizing Deep Learning to Enable Crowdsourced Development of Automated, Scalable Solutions for Documenting North American Bat Populations". In: *Journal of Applied Ecology* 59.11, pages 2849–2862. ISSN: 1365-2664. DOI: 10.1111/1365-2664.14280.

Kingma, D. P. & J. Ba (2017). "Adam: A Method for Stochastic Optimization". arXiv: 1412.6980 [cs].

Kitzes, J., R. Blake, S. Bombaci, M. Chapman, S. M. Duran, T. Huang, M. B. Joseph, S. Lapp, S. Marconi, W. K. Oestreich, T. A. Rhinehart, A. K. Schweiger, Y. Song, T. Surasinghe, D. Yang & K. Yule (2021). "Expanding NEON Biodiversity Surveys with New Instrumentation and Machine Learning Approaches". In: *Ecosphere* 12.11. DOI: 10.1002/ecs2.3795.

Knight, E., K. Hannah, G. Foley, C. Scott, R. Brigham & E. Bayne (2017). "Recommendations for Acoustic Recognizer Performance Assessment with Application to Five Common Automated Signal Recognition Programs". In: *Avian Conservation and Ecology* 12.2. ISSN: 1712-6568. DOI: 10.5751/ACE-01114-120214.

Knight, E. C. & E. M. Bayne (2018). "Classification Threshold and Training Data Affect the Quality and Utility of Focal Species Data Processed with Automated Audio-Recognition Software". In: *Bioacoustics* 28.6, pages 539–554. ISSN: 0952-4622. DOI: 10.1080/09524622.2018.1503971.

Knight, E. C., P. Sòlymos, C. Scott & E. M. Bayne (2020). "Validation Prediction: A Flexible Protocol to Increase Efficiency of Automated Acoustic Processing for Wildlife Research". In: *Ecological Applications* 30.7. ISSN: 1051-0761. DOI: 10.1002/eap.2140.

Kobayashi, K., K. Masuda, C. Haga, T. Matsui, D. Fukui & T. Machimura (2021). "Development of a Species Identification System of Japanese Bats from Echolocation Calls Using Convolutional Neural Networks". In: *Ecological Informatics* 62, page 101253. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2021.101253.

Kong, Q., Y. Cao, T. Iqbal, Y. Wang, W. Wang & M. D. Plumbley (2020). "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition". arXiv: 1912.10211 [cs, eess].

Kong, Q., C. Yu, Y. Xu, T. Iqbal, W. Wang & M. D. Plumbley (2019). "Weakly Labelled AudioSet Tagging With Attention Neural Networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.11, pages 1791–1802. ISSN: 2329-9304. DOI: 10.1109/TASLP.2019.2930913.

Kowarski, K. A. & H. Moors-Murphy (2020). "A Review of Big Data Analysis Methods for Baleen Whale Passive Acoustic Monitoring". In: *Marine Mammal Science* 37.2, pages 652–673. ISSN: 1748-7692. DOI: 10.1111/mms.12758.

Kshirsagar, M., C. Robinson, S. Yang, S. Gholami, I. Klyuzhin, S. Mukherjee, M. Nasir, A. Ortiz, F. Oviedo, D. Tanner, A. Trivedi, Y. Xu, M. Zhong, B. Dilkina, R. Dodhia & J. M. Lavista Ferres (2021). "Becoming Good at AI for Good". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 664–673. DOI: 10.1145/3461702.3462599.

Kumar, A. & B. Raj (2016). *Audio Event Detection Using Weakly Labeled Data*. DOI: 10.48550/arXiv.1605.02401. arXiv: 1605.02401. URL: http://arxiv.org/abs/1605.02401 (visited on 28/11/2024). Pre-published.

Kvsn, R. R., J. Montgomery, S. Garg & M. Charleston (2020). "Bioacoustics Data Analysis – A Taxonomy, Survey and Open Challenges". In: *IEEE Access* 8, pages 57684–57708. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2978547.

Lamont, T. A. C., L. Chapuis, B. Williams, S. Dines, T. Gridley, G. Frainer, J. Fearey, P. B. Maulana, M. E. Prasetya, J. Jompa, D. J. Smith & S. D. Simpson (2022). "HydroMoth: Testing a Prototype Low-Cost Acoustic Recorder for Aquatic Environments". In: *Remote Sensing in Ecology and Conservation* 8.3, pages 362–378. ISSN: 2056-3485. DOI: 10.1002/rse2.249.

Lamont, T. A. C., B. Williams, L. Chapuis, M. E. Prasetya, M. J. Seraphim, H. R. Harding, E. B. May, N. Janetski, J. Jompa, D. J. Smith, A. N. Radford & S. D. Simpson (2021). "The Sound of Recovery: Coral Reef Restoration Success Is

Detectable in the Soundscape". In: *Journal of Applied Ecology* 59.3, pages 742–756. DOI: 10.1111/1365-2664.14089.

Lapp, S., T. Rhinehart, L. Freeland-Haynes, J. Khilnani, A. Syunkova & J. Kitzes (2023). "OpenSoundscape: An Open-Source Bioacoustics Analysis Package for Python". In: *Methods in Ecology and Evolution* 14.9, pages 2321–2328. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14196.

Lapp, S., N. Stahlman & J. Kitzes (2023). "A Quantitative Evaluation of the Performance of the Low-Cost AudioMoth Acoustic Recording Unit". In: *Sensors* 23.11 (11), page 5254. ISSN: 1424-8220. DOI: 10.3390/s23115254.

Lapp, S., T. Wu, C. Richards-Zawacki, J. Voyles, K. M. Rodriguez, H. Shamon & J. Kitzes (2021). "Automated Detection of Frog Calls and Choruses by Pulse Repetition Rate". In: *Conservation Biology* 35.5, pages 1659–1668. DOI: 10.1111/cobi.13718. pmid: 33586273.

Lauha, P., P. Somervuo, P. Lehikoinen, L. Geres, T. Richter, S. Seibold & O. Ovaskainen (2022). "Domain-Specific Neural Networks Improve Automated Bird Sound Recognition Already with Small Amount of Local Data". In: *Methods in Ecology and Evolution* 13.12, pages 2799–2810. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14003.

Law, H. & J. Deng (2018). "Cornernet: Detecting Objects as Paired Keypoints". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750.

LeBien, J., M. Zhong, M. Campos-Cerqueira, J. P. Velev, R. Dodhia, J. L. Ferres & T. M. Aide (2020). "A Pipeline for Identification of Bird and Frog Species in Tropical Soundscape Recordings Using a Convolutional Neural Network". In: *Ecological Informatics* 59, page 101113. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2020.101113.

LeCun, Y., Y. Bengio & G. Hinton (2015). "Deep Learning". In: *Nature* 521.7553, pages 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

Leseberg, N. P., W. N. Venables, S. A. Murphy & J. E. M. Watson (2020). "Using Intrinsic and Contextual Information Associated with Automated Signal De-

tections to Improve Call Recognizer Performance: A Case Study Using the Cryptic and Critically Endangered Night Parrot Pezoporus Occidentalis". In: *Methods in Ecology and Evolution* 11.11, pages 1520–1530. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13475.

Lesmeister, D. B. & J. M. A. Jenkins (2022). "Integrating New Technologies to Broaden the Scope of Northern Spotted Owl Monitoring and Linkage with USDA Forest Inventory Data". In: *Frontiers in Forests and Global Change* 5, page 966978. DOI: 10.3389/ffgc.2022.966978.

Li, S., L. D. Xu & S. Zhao (2015). "The Internet of Things: A Survey". In: *Information systems frontiers* 17, pages 243–259.

Likens, G. & D. Lindenmayer (2018). *Effective Ecological Monitoring*. Csiro Publishing. 225 pages. ISBN: 978-1-4863-0893-4. Google Books: QDBZDwAAQBAJ.

Lin, T.-Y., P. Goyal, R. Girshick, K. He & P. Dollar (2017). "Focal Loss for Dense Object Detection". In: Proceedings of the IEEE International Conference on Computer Vision, pages 2980–2988.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár & C. L. Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*, pages 740–755. DOI: 10.1007/978-3-319-10602-1_48.

Lint, J. (1999). *Northern Spotted Owl Effectiveness Monitoring Plan for the Northwest Forest Plan*. Volume 440. US Department of Agriculture, Forest Service, Pacific Northwest Research Station.

Liu, R., J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev & J. Yosinski (2018). "An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution". In: *Advances in Neural Information Processing Systems*. Volume 31. Curran Associates, Inc.

Liu, S., A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu & B. W. Schuller (2022). "Audio Self-Supervised Learning: A Survey". In: *Patterns* 3.12, page 100616. ISSN: 2666-3899. DOI: 10.1016/j.patter.2022.100616.

Lostanlen, V., A. Bernabeu, J.-L. Béchennec, M. Briday, S. Faucou & M. Lagrange (2021). "Energy Efficiency Is Not Enough:Towards a Batteryless Internet of Sounds". In: *Proceedings of the 16th International Audio Mostly Conference*. AM '21. New York, NY, USA: Association for Computing Machinery, pages 147–155. ISBN: 978-1-4503-8569-5. DOI: 10.1145/3478384.3478408.

Lostanlen, V., J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling & J. P. Bello (2019). "Per-Channel Energy Normalization: Why and How". In: *IEEE Signal Processing Letters* 26.1, pages 39–43. ISSN: 1558-2361. DOI: 10.1109/LSP.2018.2878620.

Lostanlen, V., J. Salamon, A. Farnsworth, S. Kelling & J. P. Bello (2018). "Birdvox-Full-Night: A Dataset and Benchmark for Avian Flight Call Detection". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 266–270. DOI: 10.1109/ICASSP.2018.8461410.

Mac Aodha, O., E. Cole & P. Perona (2019). "Presence-Only Geographical Priors for Fine-Grained Image Classification". In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9596–9606.

Mac Aodha, O., R. Gibb, K. E. Barlow, E. Browning, M. Firman, R. Freeman, B. Harder, L. Kinsey, G. R. Mead, S. E. Newson, I. Pandourski, S. Parsons, J. Russ, A. Szodoray-Paradi, F. Szodoray-Paradi, E. Tilova, M. Girolami, G. Brostow & K. E. Jones (2018). "Bat Detective—Deep Learning Tools for Bat Acoustic Signal Detection". In: *PLOS Computational Biology* 14.3, e1005995. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005995.

MacIsaac, J., S. Newson, A. Ashton-Butt, H. Pearce & B. Milner (2024). "Improving Acoustic Species Identification Using Data Augmentation within a Deep Learning Framework". In: *Ecological Informatics* 83, page 102851. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2024.102851.

MacSwiney G., M. C., F. M. Clarke & P. A. Racey (2008). "What You See Is Not What You Get: The Role of Ultrasonic Detectors in Increasing Inventory

Completeness in Neotropical Bat Assemblages". In: *Journal of Applied Ecology* 45.5, pages 1364–1371. ISSN: 1365-2664. DOI: 10.1111/j.1365-2664.2008.01531.x.

Madhusudhana, S., Y. Shiu, H. Klinck, E. Fleishman, X. Liu, E.-M. Nosal, T. Helble, D. Cholewiak, D. Gillespie, A. Širović & M. A. Roch (2021). "Improve Automatic Detection of Animal Call Sequences with Temporal Context". In: *Journal of The Royal Society Interface* 18.180, page 20210297. DOI: 10.1098/rsif.2021.0297.

Marsland, S., N. Priyadarshani, J. Juodakis & I. Castro (2019). "AviaNZ: A Future-proofed Program for Annotation and Recognition of Animal Sounds in Long-time Field Recordings". In: *Methods in Ecology and Evolution* 10.8, pages 1189–1195. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13213.

Martin, K., O. Adam, N. Obin & V. Dufour (2022). "Rookognise: Acoustic Detection and Identification of Individual Rooks in Field Recordings Using Multi-Task Neural Networks". In: *Ecological Informatics* 72, page 101818. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2022.101818.

Martinez, B., J. K. Reaser, A. Dehgan, B. Zamft, D. Baisch, C. McCormick, A. J. Giordano, R. Aicher & S. Selbe (2020). "Technology Innovation: Advancing Capacities for the Early Detection of and Rapid Response to Invasive Species". In: *Biological Invasions* 22.1, pages 75–100. ISSN: 1573-1464. DOI: 10.1007/s10530-019-02146-y.

Martinsson, J., O. Mogren, M. Sandsten & T. Virtanen (2024). *From Weak to Strong Sound Event Labels Using Adaptive Change-Point Detection and Active Learning*. arXiv: 2403.08525 [cs]. URL: http://arxiv.org/abs/2403.08525 (visited on 28/10/2024). Pre-published.

Matheson, C. A. (2014). "Inaturalist". In: *Reference Reviews* 28.8, pages 36–38.

McEwen, B., K. Soltero, S. Gutschmidt, A. Bainbridge-Smith, J. Atlas & R. Green (2024). "Active Few-Shot Learning for Rare Bioacoustic Feature Annotation". In: *Ecological Informatics* 82, page 102734. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2024.102734.

McFee, B., C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg & O. Nieto (2015). "Librosa: Audio and Music Signal Analysis in Python." In: *SciPy*, pages 18–24.

McGuire, P. (2024). *BirdNET-Pi*. Version 0.13.

Melo, I., D. Llusia, R. P. Bastos & L. Signorelli (2021). "Active or Passive Acoustic Monitoring? Assessing Methods to Track Anuran Communities in Tropical Savanna Wetlands". In: *Ecological Indicators* 132, page 108305. DOI: 10.1016/j.ecolind.2021.108305.

Menghani, G. (2023). "Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better". In: *ACM Computing Surveys* 55.12, pages 1–37. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3578938. arXiv: 2106.08962 [cs].

Mesaros, A., S. Adavanne, A. Politis, T. Heittola & T. Virtanen (2019). "Joint Measurement of Localization and Detection of Sound Events". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 333–337. DOI: 10.1109/waspaa.2019.8937220.

Mesaros, A., T. Heittola, T. Virtanen & M. D. Plumbley (2021). "Sound Event Detection: A Tutorial". In: *IEEE Signal Processing Magazine* 38.5, pages 67–83. ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2021.3090678. arXiv: 2107.05463.

Metcalf, O., C. Abrahams, B. Ashington, E. Baker, T. Bradfer-Lawrence, E. Browning, J. Carruthers-Jones, J. Darby, J. Dick, A. Eldridge, D. Elliott, B. Heath, P. Howden-Leach, A. Johnston, A. Lees, C. Meyer, U. Ruiz Arana & S. Smyth (2023). *Good Practice Guidelines for Long-Term Ecoacoustic Monitoring in the UK*. Report. The UK Acoustics Network, pages 1–82. 82 pages.

Miao, Z., B. Elizalde, S. Deshmukh, J. Kitzes, H. Wang, R. Dodhia & J. M. L. Ferres (2023). *Zero-Shot Transfer for Wildlife Bioacoustics Detection*. DOI: 10.21203/rs.3.rs-3180218/v1. URL: https://www.researchsquare.com/article/rs-3180218/v1 (visited on 05/12/2024). Pre-published.

Mike Bayer (2023). *SQLAlchemy*. Version 2.0.35.

Milchram, M., M. Suarez-Rubio, A. Schröder & A. Bruckner (2020). "Estimating Population Density of Insectivorous Bats Based on Stationary Acoustic Detectors: A Case Study". In: *Ecology and Evolution* 10.3, pages 1135–1144. ISSN: 2045-7758. DOI: 10.1002/ece3.5928.

Millar, J., S. Sethi, H. Haddadi & A. Madhavapeddy (2024). *Terracorder: Sense Long and Prosper*. DOI: 10.48550/arXiv.2408.02407. arXiv: 2408.02407 [cs]. URL: http://arxiv.org/abs/2408.02407 (visited on 05/12/2024). Pre-published.

Montauban, C., M. Mas, C. Tuneu-Corral, O. S. Wangensteen, I. Budinski, J. Martí-Carreras, C. Flaquer, X. Puig-Montserrat & A. López-Baucells (2021). "Bat Echolocation Plasticity in Allopatry: A Call for Caution in Acoustic Identification of Pipistrellus Sp." In: *Behavioral Ecology and Sociobiology* 75.4, page 70. ISSN: 1432-0762. DOI: 10.1007/s00265-021-03002-7.

Mooney, T. A., L. Di Iorio, M. Lammers, T.-H. Lin, S. L. Nedelec, M. Parsons, C. Radford, E. Urban & J. Stanley (2020). "Listening Forward: Approaching Marine Biodiversity Assessments Using Acoustic Methods". In: *Royal Society Open Science* 7.8, page 201287. DOI: 10.1098/rsos.201287.

Morfi, V., Y. Bas, H. Pamuła, H. Glotin & D. Stowell (2019). "NIPS4Bplus: A Richly Annotated Birdsong Audio Dataset". In: *PeerJ Computer Science* 5, e223. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.223.

Morfi, V. & D. Stowell (2018). "Deep Learning for Audio Event Detection and Tagging on Low-Resource Datasets". In: *Applied Sciences* 8.8 (8), page 1397. ISSN: 2076-3417. DOI: 10.3390/app8081397.

Muller, M., I. Lange, D. Wang, D. Piorkowski, J. Tsay, Q. V. Liao, C. Dugan & T. Erickson (2019). "How Data Science Workers Work with Data". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 126. DOI: 10.1145/3290605.3300356.

Muller, M., C. T. Wolf, J. Andres, M. Desmond, N. N. Joshi, Z. Ashktorab, A. Sharma, K. Brimijoin, Q. Pan, E. Duesterwald & C. Dugan (2021). "Designing Ground Truth and the Social Life of Labels". In: *Proceedings of the 2021 CHI*

Conference on Human Factors in Computing Systems. DOI: 10.1145/3411764.3445402.

Nahar, N., S. Zhou, G. Lewis & C. Kästner (2022). "Collaboration Challenges in Building Ml-Enabled Systems: Communication, Documentation, Engineering, and Process". In: *Proceedings of the 44th International Conference on Software Engineering*, pages 413–425. DOI: 10.1145/3510003.3510209.

Nair, V. & G. E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

Navine, A. K., T. Denton, M. J. Weldy & P. J. Hart (2024). "All Thresholds Barred: Direct Estimation of Call Density in Bioacoustic Data". In: *Frontiers in Bird Science* 3. ISSN: 2813-3870. DOI: 10.3389/fbirs.2024.1380636.

Neves, M. & J. Seva (2020). "Annotationsaurus: A Searchable Directory of Annotation Tools". In: *arXiv*. DOI: 10.48550/arxiv.2010.06251.

Newbold, T. (2018). "Future Effects of Climate and Land-Use Change on Terrestrial Vertebrate Community Diversity under Different Scenarios". In: *Proceedings of the Royal Society B: Biological Sciences* 285.1881, page 20180792. DOI: 10.1098/rspb.2018.0792.

Newbold, T., P. Oppenheimer, A. Etard & J. J. Williams (2020). "Tropical and Mediterranean Biodiversity Is Disproportionately Sensitive to Land-Use and Climate Change". In: *Nature Ecology & Evolution* 4.12, pages 1630–1638. ISSN: 2397-334X. DOI: 10.1038/s41559-020-01303-0.

Newson, S. E., H. E. Evans & S. Gillings (2015). "A Novel Citizen Science Approach for Large-Scale Standardised Monitoring of Bat Activity and Distribution, Evaluated in Eastern England". In: *Biological Conservation* 191, pages 38–49. ISSN: 0006-3207. DOI: 10.1016/j.biocon.2015.06.009.

Nguyen, T. N. T., N. K. Nguyen, H. Phan, L. Pham, K. Ooi, D. L. Jones & W.-S. Gan (2021). "A General Network Architecture for Sound Event Localization and Detection Using Transfer Learning and Recurrent Neural Network". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal*

Processing (ICASSP), pages 935–939. DOI: 10.1109/icassp39728.2021.9414602.

Nguyen Hong Duc, P., M. Torterotot, F. Samaran, P. R. White, O. Gérard, O. Adam & D. Cazau (2021). "Assessing Inter-Annotator Agreement from Collaborative Annotation Campaign in Marine Bioacoustics". In: *Ecological Informatics* 61, page 101185. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2020.101185.

Nicholson, E., K. E. Watermeyer, J. A. Rowland, C. F. Sato, S. L. Stevenson, A. Andrade, T. M. Brooks, N. D. Burgess, S.-T. Cheng, H. S. Grantham, S. L. Hill, D. A. Keith, M. Maron, D. Metzke, N. J. Murray, C. R. Nelson, D. Obura, A. Plumptre, A. L. Skowno & J. E. M. Watson (2021). "Scientific Foundations for an Ecosystem Goal, Milestones and Indicators for the Post-2020 Global Biodiversity Framework". In: *Nature Ecology & Evolution* 5.10, pages 1338–1349. DOI: 10.1038/s41559-021-01538-5. pmid: 34400825.

Nieto-Mora, D., S. Rodríguez-Buritica, P. Rodríguez-Marín, J. Martínez-Vargaz & C. Isaza-Narváez (2023). "Systematic Review of Machine Learning Methods Applied to Ecoacoustics and Soundscape Monitoring". In: *Heliyon* 9.10, e20275. DOI: 10.1016/j.heliyon.2023.e20275. pmid: 37790981.

Nolasco, I., B. Ghani, S. Singh, E. Vidaña-Vila, H. Whitehead, E. Grout, M. Emmerson, F. Jensen, I. Kiskin, J. Morford, A. Strandburg-Peshkin, L. Gill, H. Pamuła, V. Lostanlen & D. Stowell (2023). *Few-Shot Bioacoustic Event Detection at the DCASE 2023 Challenge*. DOI: 10.48550/arXiv.2306.09223. arXiv: 2306.09223 [cs]. URL: http://arxiv.org/abs/2306.09223 (visited on 04/12/2024). Pre-published.

Nolasco, I., S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, F. H. Jensen, J. Morford, M. G. Emmerson, E. Versace, E. Grout, H. Liu, B. Ghani & D. Stowell (2023). "Learning to Detect an Animal Sound from Five Examples". In: *Ecological Informatics* 77, page 102258. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2023.102258.

Nolasco, I., S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, L. Gill, H. Pamula, V. Lostanlen, V. Morfi & D. Stowell (2022). "Few-Shot Bioacoustic Event Detection at the DCASE 2022 Challenge". In: *Workshop on Detection and Classification of Acoustic Scenes and Events*. DOI: 10.48550/arxiv.2207.07911.

Nyhus, P. J. (2016). "Human–Wildlife Conflict and Coexistence". In: *Annual Review of Environment and Resources* 41 (Volume 41, 2016), pages 143–171. ISSN: 1543-5938, 1545-2050. DOI: 10.1146/annurev-environ-110615-085634.

Obrist, M. & R. Boesch (2018). "BatScope Manages Acoustic Recordings, Analyses Calls, and Classifies Bat Species Automatically". In: *Canadian Journal of Zoology* 96.9, pages 939–954. DOI: 10.1139/cjz-2017-0103.

Odom, K. J., M. Araya-Salas, J. L. Morano, R. A. Ligon, G. M. Leighton, C. C. Taff, A. H. Dalziell, A. C. Billings, R. R. Germain, M. Pardo, L. G. de Andrade, D. Hedwig, S. C. Keen, Y. Shiu, R. A. Charif, M. S. Webster & A. N. Rice (2021). "Comparative Bioacoustics: A Roadmap for Quantifying and Comparing Animal Sounds across Diverse Taxa". In: *Biological Reviews* 96.4, pages 1135–1159. DOI: 10.1111/brv.12695. pmid: 33652499.

Oswald, J. N., A. M. Van Cise, A. Dassow, T. Elliott, M. T. Johnson, A. Ravignani & J. Podos (2022). "A Collection of Best Practices for the Collection and Analysis of Bioacoustic Data". In: *Applied Sciences* 12.23 (23), page 12046. ISSN: 2076-3417. DOI: 10.3390/app122312046.

Pantazis, O., P. Bevan, H. Pringle, G. B. Ferreira, D. J. Ingram, E. Madsen, L. Thomas, D. R. Thanet, T. Silwal, S. Rayamajhi, G. Brostow, O. M. Aodha & K. E. Jones (2024). *Deep Learning-Based Ecological Analysis of Camera Trap Images Is Impacted by Training Data Quality and Size*. DOI: 10.48550/arXiv.2408.14348. arXiv: 2408.14348 [cs]. URL: http://arxiv.org/abs/2408.14348 (visited on 28/12/2024). Pre-published.

Park, D. S., W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk & Q. V. Le (2019). *SpecAugment: A Simple Data Augmentation Method for Automatic*

Speech Recognition. DOI: 10.48550/arXiv.1904.08779. arXiv: 1904.08779. URL: http://arxiv.org/abs/1904.08779 (visited on 14/11/2024). Pre-published.

Parsons, S. & G. Jones (2000). "Acoustic Identification of Twelve Species of Echolocating Bat By Discriminant Function Analysis and Artificial Neural Networks". In: *Journal of Experimental Biology* 203.17, pages 2641–2656. ISSN: 0022-0949. DOI: 10.1242/jeb.203.17.2641.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai & S. Chintala (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pages 8024–8035.

Paullada, A., I. D. Raji, E. M. Bender, E. Denton & A. Hanna (2021). "Data and Its (Dis)Contents: A Survey of Dataset Development and Use in Machine Learning Research". In: *Patterns* 2.11, page 100336. DOI: 10.1016/j.patter.2021.100336. pmid: 34820643.

Paumen, Y., M. Mälzer, S. Alipek, J. Moll, B. Lüdtke & H. Schauer-Weisshahn (2021). "Development and Test of a Bat Calls Detection and Classification Method Based on Convolutional Neural Networks". In: *Bioacoustics* 31.5, pages 505–516. ISSN: 0952-4622. DOI: 10.1080/09524622.2021.1978863.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot & É. Duchesnay (2011). "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85, pages 2825–2830. ISSN: 1533-7928.

Pérez-Granados, C. (2023). "A First Assessment of Birdnet Performance at Varying Distances: A Playback Experiment". In: *Ardeola* 70.2, pages 257–269. ISSN: 0570-7358, 2341-0825. DOI: 10.13157/arla.70.2.2023.sc1.

Pérez-Granados, C. (2023). "BirdNET: Applications, Performance, Pitfalls and Future Opportunities". In: *Ibis* 165.3, pages 1068–1075. ISSN: 0019-1019, 1474-919X. DOI: 10.1111/ibi.13193.

Pérez-Granados, C. & J. Traba (2021). "Estimating Bird Density Using Passive Acoustic Monitoring: A Review of Methods and Suggestions for Further Research". In: *Ibis* 163.3, pages 765–783. ISSN: 1474-919X. DOI: 10.1111/ibi.12944.

Pham, P., J. Li, J. Szurley & S. Das (2018). "Eventness: Object Detection on Spectrograms for Temporal Localization of Audio Events". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2491–2495. DOI: 10.1109/ICASSP.2018.8462062.

Pichler, M. & F. Hartig (2023). "Machine Learning and Deep Learning—A Review for Ecologists". In: *Methods in Ecology and Evolution* 14.4, pages 994–1016. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14061.

Pijanowski, B. C., L. J. Villanueva-Rivera, S. L. Dumyahn, A. Farina, B. L. Krause, B. M. Napoletano, S. H. Gage & N. Pieretti (2011). "Soundscape Ecology: The Science of Sound in the Landscape". In: *BioScience* 61.3, pages 203–216. ISSN: 0006-3568. DOI: 10.1525/bio.2011.61.3.6.

Prat, Y., M. Taub & Y. Yovel (2016). "Everyday Bat Vocalizations Contain Information about Emitter, Addressee, Context, and Behavior". In: *Scientific Reports* 6.1, page 39419. ISSN: 2045-2322. DOI: 10.1038/srep39419.

Pritchard, R., L. A. Sauls, J. A. Oldekop, W. A. Kiwango & D. Brockington (2022). "Data Justice and Biodiversity Conservation". In: *Conservation Biology* 36.5, e13919. ISSN: 1523-1739. DOI: 10.1111/cobi.13919.

Rabanser, S., S. Günnemann & Z. C. Lipton (2019). *Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift*. DOI: 10.48550/arXiv.1810.11953. arXiv: 1810.11953 [stat]. URL: http://arxiv.org/abs/1810.11953 (visited on 28/11/2024). Pre-published.

Ramirez, S. (2024). *FastAPI*. Version 0.115.1.

Raschka, S. (2020). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. DOI: 10.48550/arXiv.1811.12808. arXiv: 1811.12808. URL: http://arxiv.org/abs/1811.12808 (visited on 13/11/2024). Pre-published.

Rauch, L., R. Schwinger, M. Wirth, R. Heinrich, D. Huseljic, M. Herde, J. Lange, S. Kahl, B. Sick, S. Tomforde & C. Scholz (2024). *BirdSet: A Large-Scale Dataset for Audio Classification in Avian Bioacoustics*. DOI: 10.48550/arXiv.2403.10380. arXiv: 2403.10380. URL: http://arxiv.org/abs/2403.10380 (visited on 28/11/2024). Pre-published.

Ravanelli, M. & Y. Bengio (2018). "Speaker Recognition from Raw Waveform with SincNet". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. 2018 IEEE Spoken Language Technology Workshop (SLT), pages 1021–1028. DOI: 10.1109/SLT.2018.8639585.

Reichert, B. E., M. Bayless, T. L. Cheng, J. T. H. Coleman, C. M. Francis, W. F. Frick, B. S. Gotthold, K. M. Irvine, C. Lausen, H. Li, S. C. Loeb, J. D. Reichard, T. J. Rodhouse, J. L. Segers, J. L. Siemers, W. E. Thogmartin & T. J. Weller (2021). "NABat: A Top-down, Bottom-up Solution to Collaborative Continental-Scale Monitoring". In: *Ambio* 50.4, pages 901–913. ISSN: 1654-7209. DOI: 10.1007/s13280-020-01411-y. pmid: 33454913.

Rhinehart, T. (2023). "Bioacoustics Software". In.

Rhinehart, T. A., L. M. Chronister, T. Devlin & J. Kitzes (2020). "Acoustic Localization of Terrestrial Wildlife: Current Practices and Future Opportunities". In: *Ecology and Evolution* 10.13, pages 6794–6818. ISSN: 2045-7758. DOI: 10.1002/ece3.6216.

Rhinehart, T. A., D. Turek & J. Kitzes (2022). "A Continuous-score Occupancy Model That Incorporates Uncertain Machine Learning Output from Autonomous Biodiversity Surveys". In: *Methods in Ecology and Evolution* 13.8, pages 1778–1789. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13905.

Richardson, S., A. C. Mill, D. Davis, D. Jam & A. I. Ward (2020). "A Systematic Review of Adaptive Wildlife Management for the Control of Invasive, Non-

native Mammals, and Other Human–Wildlife Conflicts". In: *Mammal Review* 50.2, pages 147–156. ISSN: 1365-2907. DOI: 10.1111/mam.12182.

Riede, K. & R. Balakrishnan (2024). "Acoustic Monitoring for Tropical Insect Conservation". In: *bioRxiv*. DOI: 10.1101/2024.07.03.601657.

Robinson, D., A. Robinson & L. Akrapongpisak (2023). "Transferable Models for Bioacoustics with Human Language Supervision". In: *arXiv.org*. DOI: 10.48550/arxiv.2308.04978.

Robinson, J. M., M. F. Breed & C. Abrahams (2023). "The Sound of Restored Soil: Measuring Soil Biodiversity in a Forest Restoration Chronosequence with Ecoacoustics". In: *bioRxiv*. DOI: 10.1101/2023.01.23.525240.

Roch, M. A., H. Batchelor, S. Baumann-Pickering, C. L. Berchok, D. Cholewiak, E. Fujioka, E. C. Garland, S. Herbert, J. A. Hildebrand, E. M. Oleson, S. Van Parijs, D. Risch, A. Širović & M. S. Soldevilla (2016). "Management of Acoustic Metadata for Bioacoustics". In: *Ecological Informatics* 31, pages 122–136. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2015.12.002.

Roe, P., P. Eichinski, R. A. Fuller, P. G. McDonald, L. Schwarzkopf, M. Towsey, A. Truskinger, D. Tucker & D. M. Watson (2021). "The Australian Acoustic Observatory". In: *Methods in Ecology and Evolution* 12.10, pages 1802–1808. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13660.

Roemer, C., J.-F. Julien, P. P. Ahoudji, J.-M. Chassot, M. Genta, R. Colombo, G. Botto, C. A. Negreira, B. A. Djossa, R. K. Ing, A. Hassanin, V. Rufray, Q. Uriot, V.-C. Participants & Y. Bas (2021). "An Automatic Classifier of Bat Sonotypes around the World". In: *Methods in Ecology and Evolution* 12.12, pages 2432–2444. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13721.

Rokh, B., A. Azarpeyvand & A. Khanteymoori (2023). "A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification". In: *ACM Transactions on Intelligent Systems and Technology* 14.6, pages 1–50. ISSN: 2157-6904, 2157-6912. DOI: 10.1145/3623402. arXiv: 2205.07877 [cs].

Ronneberger, O., P. Fischer & T. Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Edited by N. Navab, J. Hornegger, W. M. Wells & A. F. Frangi. Volume 9351. Cham: Springer International Publishing, pages 234–241. ISBN: 978-3-319-24573-7 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.

Roscher, R., M. Russwurm, C. Gevaert, M. Kampffmeyer, J. A. Dos Santos, M. Vakalopoulou, R. Hänsch, S. Hansen, K. Nogueira, J. Prexl & D. Tuia (2024). "Better, Not Just More: Data-centric Machine Learning for Earth Observation". In: *IEEE Geoscience and Remote Sensing Magazine* 12.4, pages 335–355. ISSN: 2168-6831. DOI: 10.1109/MGRS.2024.3470986.

Ross, S. R. P.-J., D. P. O'Connell, J. L. Deichmann, C. Desjonquères, A. Gasc, J. N. Phillips, S. S. Sethi, C. M. Wood & Z. Burivalova (2023). "Passive Acoustic Monitoring Provides a Fresh Perspective on Fundamental Ecological Questions". In: *Functional Ecology* 37.4, pages 959–975. ISSN: 1365-2435. DOI: 10.1111/1365-2435.14275.

Ross, S. R.-J., N. R. Friedman, M. Yoshimura, T. Yoshida, I. Donohue & E. P. Economo (2021). "Utility of Acoustic Indices for Ecological Monitoring in Complex Sonic Environments". In: *Ecological Indicators* 121, page 107114. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2020.107114.

Russ, J. (2021). *Bat Calls of Britain and Europe: A Guide to Species Identification*. Pelagic Publishing Ltd.

Russo, D., L. Ancillotto & G. Jones (2018). "Bats Are Still Not Birds in the Digital Era: Echolocation Call Variation and Why It Matters for Bat Species Identification". In: *Canadian Journal of Zoology* 96.2, pages 63–78. DOI: 10.1139/cjz-2017-0089.

Russo, D., V. B. Salinas-Ramos, L. Cistrone, S. Smeraldo, L. Bosso & L. Ancillotto (2021). "Do We Need to Use Bats as Bioindicators?" In: *Biology* 10.8, page 693. ISSN: 2079-7737. DOI: 10.3390/biology10080693. pmid: 34439926.

Sager, C., C. Janiesch & P. Zschech (2021). "A Survey of Image Labelling for Computer Vision Applications". In: *Journal of Business Analytics* 4.2, pages 91–110. DOI: 10.1080/2573234x.2021.1908861.

Salamon, J., D. MacConnell, M. Cartwright, P. Li & J. P. Bello (2017). "Scaper: A Library for Soundscape Synthesis and Augmentation". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pages 344–348. DOI: 10.1109/WASPAA.2017.8170052.

Sambasivan, N., S. Kapania, H. Highfill, D. Akrong, P. Paritosh & L. M. Aroyo (2021). ""Everyone Wants to Do the Model Work, Not the Data Work": Data Cascades in High-Stakes AI". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. DOI: 10.1145/3411764.3445518.

Saria, S. & A. Subbaswamy (2019). "Tutorial: Safe and Reliable Machine Learning". In: *arXiv: Learning*.

Schmeller, D. S., R. Julliard, P. J. Bellingham, M. Böhm, N. Brummitt, A. Chiarucci, D. Couvet, S. Elmendorf, D. M. Forsyth, J. G. Moreno, R. D. Gregory, W. E. Magnusson, L. J. Martin, M. A. McGeoch, J.-B. Mihoub, H. M. Pereira, V. Proença, C. A. van Swaay, T. Yahara & J. Belnap (2015). "Towards a Global Terrestrial Species Monitoring Program". In: *Journal for Nature Conservation* 25, pages 51–57. ISSN: 1617-1381. DOI: 10.1016/j.jnc.2015.03.003.

Schwab, E., S. Pogrebnoj, M. Freund, F. Flossmann, S. Vogl & K.-H. Frommolt (2022). "Automated Bat Call Classification Using Deep Convolutional Neural Networks". In: *Bioacoustics* 32.1, pages 1–16. ISSN: 0952-4622. DOI: 10.1080/09524622.2022.2050816.

Sechidis, K., G. Tsoumakas & I. Vlahavas (2011). "On the Stratification of Multi-label Data". In: *Machine Learning and Knowledge Discovery in Databases*. Edited by D. Gunopulos, T. Hofmann, D. Malerba & M. Vazirgiannis. Volume 6913. Berlin, Heidelberg: Springer Berlin Heidelberg, pages 145–158. ISBN: 978-3-642-23807-9 978-3-642-23808-6. DOI: 10.1007/978-3-642-23808-6_10.

Secretariat, G. (2023). *GBIF Backbone Taxonomy*. DOI: 10.15468/39omei.

Sethi, S. S., A. Bick, M.-Y. Chen, R. Crouzeilles, B. V. Hillier, J. Lawson, C.-Y. Lee, S.-H. Liu, C. H. de Freitas Parruco, C. M. Rosten, M. Somveille, M.-N. Tuanmu & C. Banks-Leite (2024). "Large-Scale Avian Vocalization Detection Delivers Reliable Global Biodiversity Insights". In: *Proceedings of the National Academy of Sciences* 121.33, e2315933121. DOI: 10.1073/pnas.2315933121.

Sethi, S. S., R. M. Ewers, N. S. Jones, C. D. L. Orme & L. Picinali (2018). "Robust, Real-time and Autonomous Monitoring of Ecosystems with an Open, Low-cost, Networked Device". In: *Methods in Ecology and Evolution* 9.12, pages 2383–2387. ISSN: 2041-210X. DOI: 10.1111/2041-210x.13089.

Sethi, S. S., R. M. Ewers, N. S. Jones, A. Signorelli, L. Picinali & C. D. L. Orme (2020). "SAFE Acoustics: An Open-Source, Real-Time Eco-Acoustic Monitoring Network in the Tropical Rainforests of Borneo". In: *Methods in Ecology and Evolution* 11.10, pages 1182–1185. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13438.

Sethi, S. S., R. M. Ewers, N. S. Jones, J. Sleutel, A. Shabrani, N. Zulkifli & L. Picinali (2022). "Soundscapes Predict Species Occurrence in Tropical Forests". In: *Oikos* 2022.3, e08525. ISSN: 1600-0706. DOI: 10.1111/oik.08525.

Shah, A., A. Kumar, A. G. Hauptmann & B. Raj (2018). *A Closer Look at Weak Label Learning for Audio Events*. DOI: 10.48550/arXiv.1804.09288. arXiv: 1804.09288 [cs]. URL: http://arxiv.org/abs/1804.09288 (visited on 28/11/2024). Pre-published.

Sharma, S., K. Sato & B. P. Gautam (2022). "Bioacoustics Monitoring of Wildlife Using Artificial Intelligence: A Methodological Literature Review". In: *2022 International Conference on Networking and Network Applications (NaNA)*. 2022 International Conference on Networking and Network Applications (NaNA), pages 1–9. DOI: 10.1109/NaNA56854.2022.00063.

Sheng, Z., S. Pfersich, A. Eldridge, J. Zhou, D. Tian & V. C. M. Leung (2019). "Wireless Acoustic Sensor Networks and Edge Computing for Rapid Acoustic Monitoring". In: *IEEE/CAA Journal of Automatica Sinica* 6.1, pages 64–74. ISSN: 2329-9274. DOI: 10.1109/JAS.2019.1911324.

Shi, W., J. Cao, Q. Zhang, Y. Li & L. Xu (2016). "Edge Computing: Vision and Challenges". In: *IEEE Internet of Things Journal* 3.5, pages 637–646. ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2579198.

Simonyan, K. & A. Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". arXiv: 1409.1556 [cs].

Simpson, R., K. R. Page & D. De Roure (2014). "Zooniverse". In: *Proceedings of the 23rd International Conference on World Wide Web*, pages 1049–1054. DOI: 10.1145/2567948.2579215.

Solem, A. & A. Saif Uddin (2024). *Celery*. Version 5.4.

Somwong, B., K. Kumphet & W. Massagram (2023). "Acoustic Monitoring System with AI Threat Detection System for Forest Protection". In: *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 253–257. DOI: 10.1109/jcsse58229.2023.10202043.

Song, Y., T. Wang, P. Cai, S. K. Mondal & J. P. Sahoo (2023). "A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities". In: *ACM Computing Surveys* 55 (13s), pages 1–40. DOI: 10.1145/3582688.

Sparrow, B. D., W. Edwards, S. E. Munroe, G. M. Wardle, G. R. Guerin, J.-F. Bastin, B. Morris, R. Christensen, S. Phinn & A. J. Lowe (2020). "Effective Ecosystem Monitoring Requires a Multi-scaled Approach". In: *Biological Reviews* 95.6, pages 1706–1719. DOI: 10.1111/brv.12636. pmid: 32648358.

Stähli, O., T. Ost & T. Studer (2022). "Development of an AI-based Bioacoustic Wolf Monitoring System". In: *The International FLAIRS Conference Proceedings* 35. DOI: 10.32473/flairs.v35i.130552.

Standley, T., A. R. Zamir, D. Chen, L. Guibas, J. Malik & S. Savarese (2020). *Which Tasks Should Be Learned Together in Multi-task Learning?* DOI: 10.48550/arXiv.1905.07553. arXiv: 1905.07553 [cs]. URL: http://arxiv.org/abs/1905.07553 (visited on 16/06/2023). Pre-published.

Stephenson, P. J., M. C. Londoño-Murcia, P. A. V. Borges, L. Claassens, H. Frisch-Nwakanma, N. Ling, S. McMullan-Fisher, J. J. Meeuwig, K. M. M. Unter, J. L.

Walls, I. J. Burfield, D. do Carmo Vieira Correa, G. N. Geller, I. Montenegro Paredes, L. K. Mubalama, Y. Ntiamoa-Baidu, I. Roesler, F. Rovero, Y. P. Sharma, N. W. Wiwardhana, J. Yang & L. Fumagalli (2022). "Measuring the Impact of Conservation: The Growing Importance of Monitoring Fauna, Flora and Funga". In: *Diversity* 14.10, page 824. DOI: 10.3390/d14100824.

Stowell, D. (2022). "Computational Bioacoustics with Deep Learning: A Review and Roadmap". In: *PeerJ* 10, e13152. ISSN: 2167-8359. DOI: 10.7717/peerj.13152.

Stretcu, O., E. Vendrow, K. Hata, K. Viswanathan, V. Ferrari, S. Tavakkol, W. Zhou, A. Avinash, E. Luo, N. G. Alldrin, M. Bateni, G. Berger, A. Bunner, C.-T. Lu, J. Rey, G. DeSalvo, R. Krishna & A. Fuxman (2023). "Agile Modeling: From Concept to Classifier in Minutes". In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 22323–22334.

Sugai, L. S. M., T. S. F. Silva, J. W. Ribeiro & D. Llusia (2018). "Terrestrial Passive Acoustic Monitoring: Review and Perspectives". In: *BioScience* 69.1, pages 15–25. ISSN: 0006-3568. DOI: 10.1093/biosci/biy147.

Surianarayanan, C., J. J. Lawrence, P. R. Chelliah, E. Prakash & C. Hewage (2023). "A Survey on Optimization Techniques for Edge Artificial Intelligence (AI)". In: *Sensors* 23.3 (3), page 1279. ISSN: 1424-8220. DOI: 10.3390/s23031279.

Szewczak, J. M. (2010). *SonoBat*. Version 3.

Szewczak, J. (2004). "Advanced Analysis Techniques for Identifying Bat Species". In: *Bat echolocation research: tools, techniques and analysis. Bat Conservation International, Austin, Texas, USA*, pages 121–126.

Tabak, M. A., K. L. Murray, A. M. Reed, J. A. Lombardi & K. J. Bay (2022). "Automated Classification of Bat Echolocation Call Recordings with Artificial Intelligence". In: *Ecological Informatics* 68, page 101526. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2021.101526.

TDWG (2023). *Audiovisual Core Introduction*. Biodiversity Information Standards (TDWG).

Teixeira, D., M. Maron & B. J. van Rensburg (2019). "Bioacoustic Monitoring of Animal Vocal Behavior for Conservation". In: *Conservation Science and Practice* 1.8. DOI: 10.1111/csp2.72.

Teixeira, D., P. Roe, B. J. van Rensburg, S. Linke, P. G. McDonald, D. Tucker & S. Fuller (2024). "Effective Ecological Monitoring Using Passive Acoustic Sensors: Recommendations for Conservation Practitioners". In: *Conservation Science and Practice* 6.6, e13132. ISSN: 2578-4854. DOI: 10.1111/csp2.13132.

Tejero, J. G., M. S. Zinkernagel, S. Wolf, R. Sznitman & P. Márquez-Neila (2023). "Full or Weak Annotations? An Adaptive Strategy for Budget-Constrained Annotation Campaigns". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11381–11391.

Tenopir, C., N. M. Rice, S. Allard, L. Baird, J. Borycz, L. Christian, B. Grant, R. Olendorf & R. J. Sandusky (2020). "Data Sharing, Management, Use, and Reuse: Practices and Perceptions of Scientists Worldwide". In: *PLOS ONE* 15.3, e0229003. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0229003.

Tilman, D., M. Clark, D. R. Williams, K. Kimmel, S. Polasky & C. Packer (2017). "Future Threats to Biodiversity and Pathways to Their Prevention". In: *Nature* 546.7656, pages 73–81. ISSN: 1476-4687. DOI: 10.1038/nature22900. pmid: 28569796.

Tilman, D., F. Isbell & J. M. Cowles (2014). "Biodiversity and Ecosystem Functioning". In: *Annual Review of Ecology, Evolution, and Systematics* 45.1, pages 471–493. DOI: 10.1146/annurev-ecolsys-120213-091917.

Tkachenko, M., M. Malyuk, A. Holmanyuk & N. Liubimov (2020–2022). *Label Studio: Data Labeling Software*.

Tsalera, E., A. Papadakis & M. Samarakou (2021). "Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning". In: *Journal of Sensor and Actuator Networks* 10.4 (4), page 72. ISSN: 2224-2708. DOI: 10.3390/jsan10040072.

Tuia, D., B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski,

I. D. Couzin, G. van Horn, M. C. Crofoot, C. V. Stewart & T. Berger-Wolf (2022). "Perspectives in Machine Learning for Wildlife Conservation". In: *Nature Communications* 13.1 (1), page 792. ISSN: 2041-1723. DOI: 10.1038/s41467-022-27980-y.

Ulloa, J. S., S. Haupert, J. F. Latorre, T. Aubin & J. Sueur (2021). "Scikit-maad: An Open-source and Modular Toolbox for Quantitative Soundscape Analysis in Python". In: *Methods in Ecology and Evolution* 12.12, pages 2334–2340. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13711.

Van Merriënboer, B., J. Hamer, V. Dumoulin, E. Triantafillou & T. Denton (2024). "Birds, Bats and beyond: Evaluating Generalization in Bioacoustics Models". In: *Frontiers in Bird Science* 3. ISSN: 2813-3870. DOI: 10.3389/fbirs.2024.1369756.

Van Rossum, G. & F. L. Drake Jr (1995). *Python Tutorial*. Volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Vaswani, A. (2017). "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*.

Vellinga, W.-P. & R. Planque (2015). "The Xeno-canto Collection and Its Relation to Sound Recognition and Classification". In: *CLEF (Working Notes)*, page 11.

Venkatesh, S., D. Moffat & E. R. Miranda (2022). "You Only Hear Once: A YOLO-like Algorithm for Audio Segmentation and Sound Event Detection". In: *Applied Sciences* 12.7 (7), page 3293. ISSN: 2076-3417. DOI: 10.3390/app12073293.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa & P. van Mulbregt (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (3), pages 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2.

Vogelbacher, M., H. Bellafkir, J. Gottwald, D. Schneider, M. Muhling & B. Freisleben (2023). "Deep Learning for Recognizing Bat Species and Bat Behavior in Audio Recordings". In: *2023 10th IEEE Swiss Conference on Data Science (SDS)*. 2023 10th IEEE Swiss Conference on Data Science (SDS), pages 50–57. DOI: 10.1109/SDS57534.2023.00014.

Walke, J. (2023). *React*. Version 18.

Walters, C. L., A. Collen, T. Lucas, K. Mroz, C. A. Sayer & K. E. Jones (2013). "Challenges of Using Bioacoustics to Globally Monitor Bats". In: *Bat Evolution, Ecology, and Conservation*. Edited by R. A. Adams & S. C. Pedersen. New York, NY: Springer, pages 479–499. ISBN: 978-1-4614-7397-8. DOI: 10.1007/978-1-4614-7397-8_23.

Wang, Y., P. Getreuer, T. Hughes, R. F. Lyon & R. A. Saurous (2017). "Trainable Frontend for Robust and Far-Field Keyword Spotting". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5670–5674. DOI: 10.1109/ICASSP.2017.7953242.

Wang, Y., M. Cartwright & J. P. Bello (2022). "Active Few-Shot Learning for Sound Event Detection". In: Proc. Interspeech 2022, pages 1551–1555. DOI: 10.21437/Interspeech.2022-10907.

Wang, Y., J. Ye, X. Li & D. L. Borchers (2024). "Towards Automated Animal Density Estimation with Acoustic Spatial Capture-Recapture". In: *Biometrics* 80.3, ujae081. ISSN: 0006-341X. DOI: 10.1093/biomtc/ujae081.

Wei, S., S. Zou, F. Liao & w. lang (2020). "A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification". In: *Journal of Physics: Conference Series* 1453.1, page 012085. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1453/1/012085.

Weldy, M. J., D. B. Lesmeister, C. B. Yackulic, C. L. Appel, C. McCafferty & J. David Wiens (2023). "Long-Term Monitoring in Transition: Resolving Spatial Mismatch and Integrating Multistate Occupancy Data". In: *Ecological Indicators* 146, page 109815. DOI: 10.1016/j.ecolind.2022.109815.

Whytock, R. C. & J. Christie (2016). "Solo: An Open Source, Customizable and Inexpensive Audio Recorder for Bioacoustic Research". In: *Methods in Ecology and Evolution* 8.3, pages 308–312. DOI: 10.1111/2041-210x.12678.

Wieczorek, J., D. Bloom, R. Guralnick, S. Blum, M. Döring, R. Giovanni, T. Robertson & D. Vieglais (2012). "Darwin Core: An Evolving Community-Developed Biodiversity Data Standard". In: *PLOS ONE* 7.1, e29715. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0029715.

Wijers, M., A. Loveridge, D. W. Macdonald & A. Markham (2019). "CARACAL: A Versatile Passive Acoustic Monitoring Tool for Wildlife Research and Conservation". In: *Bioacoustics* 30.1, pages 41–57. ISSN: 0952-4622. DOI: 10.1080/09524622.2019.1685408.

Wilkinson, M. D., M. Dumontier, IJ. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao & B. Mons (2016). "The FAIR Guiding Principles for Scientific Data Management and Stewardship". In: *Sci. Data* 3.1, page 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

Williams, B., B. van Merriënboer, V. Dumoulin, J. Hamer, E. Triantafillou, A. B. Fleishman, M. McKown, J. E. Munger, A. N. Rice, A. Lillis, C. E. White, C. A. D. Hobbs, T. B. Razak, K. E. Jones & T. Denton (2024). *Leveraging Tropical Reef, Bird and Unrelated Sounds for Superior Transfer Learning in Marine Bioacoustics*. DOI: 10.48550/arXiv.2404.16436. arXiv: 2404.16436 [cs]. URL: http://arxiv.org/abs/2404.16436 (visited on 05/12/2024). Pre-published.

Williams, D. R., A. Balmford & D. S. Wilcove (2020). "The Past and Future Role of Conservation Science in Saving Biodiversity". In: *Conservation Letters* 13.4. DOI: 10.1111/conl.12720.

Williams, E. M., C. F. J. O'Donnell & D. P. Armstrong (2018). "Cost-benefit Analysis of Acoustic Recorders as a Solution to Sampling Challenges Experienced Monitoring Cryptic Species". In: *Ecology and Evolution* 8.13, pages 6839–6848. DOI: 10.1002/ece3.4199. pmid: 30038779.

Wood, C. M., F. Günther, A. Rex, D. F. Hofstadter, H. Reers, S. Kahl, M. Z. Peery & H. Klinck (2024). "Real-Time Acoustic Monitoring Facilitates the Proactive Management of Biological Invasions". In: *Biological Invasions* 26.12, pages 3989–3996. ISSN: 1573-1464. DOI: 10.1007/s10530-024-03426-y.

Wood, C. M. & S. Kahl (2024). "Guidelines for Appropriate Use of BirdNET Scores and Other Detector Outputs". In: *Journal of Ornithology* 165.3, pages 777–782. ISSN: 2193-7206. DOI: 10.1007/s10336-024-02144-5.

Wood, C. M. & M. Z. Peery (2022). "What Does 'Occupancy' Mean in Passive Acoustic Surveys?" In: *Ibis* 164.4, pages 1295–1300. ISSN: 1474-919X. DOI: 10.1111/ibi.13092.

Wrege, P. H., E. D. Rowland, S. Keen & Y. Shiu (2017). "Acoustic Monitoring for Conservation in Tropical Forests: Examples from Forest Elephants". In: *Methods in Ecology and Evolution* 8.10, pages 1292–1301. ISSN: 2041-210X. DOI: 10.1111/2041-210x.12730.

WWF (2024). *Living Planet Report 2024 – A System in Peril*. Gland, Switzerland: WWF.

Xie, J., J. G. Colonna & J. Zhang (2021). "Bioacoustic Signal Denoising: A Review". In: *Artificial Intelligence Review* 54.5, pages 3575–3597. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09932-4.

Xu, H., Y. Cao, D. Yu, M. Cao, Y. He, M. Gill & H. M. Pereira (2021). "Ensuring Effective Implementation of the Post-2020 Global Biodiversity Targets". In: *Nature Ecology & Evolution* 5.4, pages 411–418. ISSN: 2397-334X. DOI: 10.1038/s41559-020-01375-y.

Ying, X. (2019). "An Overview of Overfitting and Its Solutions". In: *Journal of Physics: Conference Series* 1168, page 022022. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1168/2/022022.

Yılmaz, B., M. Sen, E. Masazade & V. Beskardes (2022). "Behavior Classification of Egyptian Fruit Bat (Rousettus Aegyptiacus) From Calls With Deep Learning". In: *Advances in Computational Intelligence and Robotics*, pages 60–98. DOI: 10.4018/978-1-7998-8686-0.ch004.

Yoh, N., T. Kingston, E. McArthur, O. E. Aylen, J. C.-C. Huang, E. R. Jinggong, F. A. A. Khan, B. P. Lee, S. L. Mitchell, J. E. Bicknell & M. J. Struebig (2022). "A Machine Learning Framework to Classify Southeast Asian Echolocating Bats". In: *Ecological Indicators* 136, page 108696. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2022.108696.

Yoh, N., D. J. I. Seaman, N. J. Deere, H. Bernard, J. E. Bicknell & M. J. Struebig (2023). "Benign Effects of Logging on Aerial Insectivorous Bats in Southeast Asia Revealed by Remote Sensing Technologies". In: *Journal of Applied Ecology* 60.7, pages 1210–1222. ISSN: 1365-2664. DOI: 10.1111/1365-2664.14398.

Zamora-Gutierrez, V., M. C. MacSwiney G., S. Martínez Balvanera & E. Robredo Esquivelzeta (2021). "The Evolution of Acoustic Methods for the Study of Bats". In: *50 Years of Bat Research: Foundations and New Frontiers*. Edited by B. K. Lim, M. B. Fenton, R. M. Brigham, S. Mistry, A. Kurta, E. H. Gillam, A. Russell & J. Ortega. Fascinating Life Sciences. Cham: Springer International Publishing, pages 43–59. ISBN: 978-3-030-54727-1. DOI: 10.1007/978-3-030-54727-1_3.

Zamora-Gutierrez, V., C. Lopez-Gonzalez, M. C. MacSwiney Gonzalez, B. Fenton, G. Jones, E. K. V. Kalko, S. J. Puechmaille, V. Stathopoulos & K. E. Jones (2016). "Acoustic Identification of Mexican Bats Based on Taxonomic and Ecological Constraints on Call Design". In: *Methods in Ecology and Evolution* 7.9, pages 1082–1091. ISSN: 2041-210X. DOI: 10.1111/2041-210x.12556.

Zamora-Gutierrez, V., J. Ortega, R. Avila-Flores, P. A. Aguilar-Rodríguez, M. Alarcón-Montano, L. G. Avila-Torresagatón, J. Ayala-Berdón, B. Bolívar-Cimé, M. Briones-Salas, M. Chan-Noh, M. Chávez-Cauich, C. Chávez, P. Cortés-Calva, J. Cruzado, J. C. Cuevas, M. Del Real-Monroy, C. Elizalde-Arellano, M. García-Luis, R. García-Morales, J. A. Guerrero, A. A. Guevara-Carrizales, E. G. Gutiérrez, L. A. Hernández-Mijangos, M. P. Ibarra-López, L. I. Iñiguez-Dávalos, R. León-Madrazo, C. López-González, M. C. López-Téllez, J. C. López-Vidal, S. Martínez-Balvanera, F. Montiel-Reyes, R. Murrieta-Galindo, C. L. Orozco-Lugo, J. M. Pech-Canché, L. Pérez-Pérez, M. M. Ramírez-Martínez, A. Rizo-Aguilar, E. Robredo-Esquivelzeta, A. Z. Rodas-Martínez, M. A. Rojo-Cruz, C. I. Selem-Salas, E. Uribe-Bencomo, J. A. Vargas-Contreras & M. C. Mac-Swiney G. (2020). "The Sonozotz Project: Assembling an Echolocation Call Library for Bats in a Megadiverse Country". In: *Ecology and Evolution* 10.11, pages 4928–4943. ISSN: 2045-7758. DOI: 10.1002/ece3.6245.

Zha, D., Z. P. Bhat, K.-H. Lai, F. Yang & X. Hu (2023). "Data-Centric AI: Perspectives and Challenges". In: *arXiv.org*. DOI: 10.48550/arxiv.2301.04819.

Zha, D., Z. P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong & X. Hu (2023). "Data-Centric Artificial Intelligence: A Survey". In: *arXiv.org*. DOI: 10.48550/arxiv.2303.10158.

Zhang, A. X., M. Muller & D. Wang (2020). "How Do Data Science Workers Collaborate? Roles, Workflows, and Tools". In: *Proceedings of the ACM on Human-Computer Interaction* 4 (CSCW1), pages 1–23. DOI: 10.1145/3392826.

Zhang, H., M. Cisse, Y. N. Dauphin & D. Lopez-Paz (2018). "Mixup: Beyond Empirical Risk Minimization". arXiv: 1710.09412 [cs, stat].

Zhang, J., K. Huang, M. Cottman-Fields, A. Truskinger, P. Roe, S. Duan, X. Dong, M. Towsey & J. Wimmer (2013). "Managing and Analysing Big Audio Data for Environmental Monitoring". In: *2013 IEEE 16th International Conference on Computational Science and Engineering*. DOI: 10.1109/cse.2013.146.

Zhang, K., T. Liu, S. Song, X. Zhao, S. Sun, W. Metzner, J. Feng & Y. Liu (2021). "Separating Overlapping Bat Calls with a Bi-directional Long Short-

term Memory Network". In: *Integrative Zoology* 17.5, pages 741–751. ISSN: 1749-4877. DOI: 10.1111/1749-4877.12549.

Zhang, R. (2019). *Making Convolutional Networks Shift-Invariant Again*. DOI: 10.48550/arXiv.1904.11486. arXiv: 1904.11486 [cs]. URL: http://arxiv.org/abs/1904.11486 (visited on 07/01/2025). Pre-published.

Zhang, Y. & Q. Yang (2022). "A Survey on Multi-Task Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 34.12, pages 5586–5609. ISSN: 1558-2191. DOI: 10.1109/TKDE.2021.3070203.

Zhou, X., D. Wang & P. Krähenbühl (2019). "Objects as Points". arXiv: 1904.07850 [cs].

Zhuang, F., Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong & Q. He (2021). "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1, pages 43–76. DOI: 10.1109/jproc.2020.3004555.

Znidersic, E. & D. M. Watson (2022). "Acoustic Restoration: Using Soundscapes to Benchmark and Fast-track Recovery of Ecological Communities". In: *Ecology Letters* 25.7, pages 1597–1603. ISSN: 1461-0248. DOI: 10.1111/ele.14015.

Zou, Z., K. Chen, Z. Shi, Y. Guo & J. Ye (2023). "Object Detection in 20 Years: A Survey". In: *Proceedings of the IEEE* 111.3, pages 257–276. ISSN: 1558-2256. DOI: 10.1109/JPROC.2023.3238524.

Zu Ermgassen, S. O. S. E., S. Marsh, K. Ryland, E. Church, R. Marsh & J. W. Bull (2021). "Exploring the Ecological Outcomes of Mandatory Biodiversity Net Gain Using Evidence from Early-Adopter Jurisdictions in England". In: *Conservation Letters* 14.6, e12820. ISSN: 1755-263X. DOI: 10.1111/conl.12820.

Zualkernan, I., J. Judas, T. Mahbub, A. Bhagwagar & P. Chand (2020). "A Tiny CNN Architecture for Identifying Bat Species from Echolocation Calls". In: *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*. 2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G), pages 81–86. DOI: 10.1109/AI4G50087.2020.9311084.

Zualkernan, I., J. Judas, T. Mahbub, A. Bhagwagar & P. Chand (2021). "An AIoT System for Bat Species Classification". In: *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*. 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pages 155–160. DOI: 10.1109/IoTaIS50849.2021.9359704.

Zwerts, J. A., P. J. Stephenson, F. Maisels, M. Rowcliffe, C. Astaras, P. A. Jansen, J. van der Waarde, L. E. H. M. Sterck, P. A. Verweij, T. Bruce, S. Brittain & M. van Kuijk (2021). "Methods for Wildlife Monitoring in Tropical Forests: Comparing Human Observations, Camera Traps, and Passive Acoustic Sensors". In: *Conservation Science and Practice* 3.12. DOI: 10.1111/csp2.568.

Appendix A

Appendix for Chapter 2

A.1 Annotation tool comparison

To obtain a comprehensive list of potential alternative tools to compare with `whombat`, I conducted a thorough search using multiple sources. Our search strategy included three main categories of sources. First, I conducted searches in academic databases, specifically the Web of Science Core Collection and the IEEE Xplore Digital Library, for publications related to bioacoustic and audio annotation tools. Second, I used search engines such as Google and GitHub to broaden our search. Specifically, I conducted a Google search for "audio annotation tool" and "bioacoustic software" and searched on GitHub for public repositories with the tags "audio" and "annotation." Finally, I consulted compiled lists of bioacoustic and annotation software. Specifically, I referred to a GitHub repository maintained by rhine3 (`https://github.com/rhine3/bioacoustics-software`), which provides an up-to-date list of bioacoustic software (Rhinehart, 2023), a Wikipedia article that lists bioacoustic software, and a compilation of bioacoustic software by the Evergreen State College. Additionally, I explored several lists of annotation software on GitHub, including heartexlabs/awesome-data-labeling, taivop/awesome-data-annotation, and jsbroks/awesome-dataset-tools.

To narrow down the list of potential alternative tools, I established specific criteria that each tool had to meet. Firstly, the tool had to be available for installation or

use through a web service. I excluded any tool that require complex installation procedures or have outdated dependencies. Secondly, the tool had to be user-friendly, which meant that it had to have a Graphical User Interface (GUI) and require no coding skills, as our goal was to identify tools that could be used by researchers with minimal technical expertise. Thirdly, the tool had to be capable of visualising audio files, either as a waveform or a spectrogram-like representation, as this is a fundamental aspect of bioacoustic annotation. Fourthly, the tool had to provide a means for manual annotation, and therefore I only considered tools that were capable of generating annotations themselves. Several bioacoustic tools provide automatic annotation capabilities, such as SonoBat (Szewczak, 2010), but do not provide a means for manual annotation and therefore were excluded from our analysis. Finally, I excluded any services that involved hiring external annotators, such as Amazon Mechanical Turk. By applying these criteria, I was able to filter out tools that did not meet our requirements and narrow down our list of potential alternative tools to compare with `whombat`. In total, I evaluated 45 audio annotation tools.

The tools were then evaluated based on the following criteria:

Open source Whether the annotation tool is open source or not. Open-source tools allow users to access and modify the source code, which can be beneficial for researchers who need to customise the tool to fit their specific research needs.

Self-hosted Whether the tool can be self-hosted, meaning it can be installed on a local server or personal computer and used without an internet connection. This is important for researchers who need to work with sensitive data that cannot be uploaded to a cloud-based platform, or are working under limited connectivity conditions.

Collaborative Whether multiple users can use the tool at the same time. This is important for collaborative research projects where multiple annotators need to work on the same dataset simultaneously.

Large Datasets This criterion evaluates the ability of the tool to efficiently manage

large datasets, enabling users to work with collections of recordings within a single workspace. Specifically, it assesses whether the tool enables users to browse quickly through multiple recordings without the need to manually load and unload each recording. While some tools, such as Raven (Conservation Bioacoustics, 2023) and Sonic Visualiser (Cannam et al., 2010), have the capability to load multiple recordings simultaneously, they may not be optimised for analysing large datasets.

Rich Metadata Whether the tool can store and display rich metadata about the recordings. Many audio workstation tools, like Audacity (Audacity, 2017), do not display metadata about the recordings aside from the file name. Others, like Raven (Conservation Bioacoustics, 2023), can display metadata about the recordings but do not allow the user to edit the metadata.

Search Capabilities Whether the tool has search capabilities, allowing users to find specific annotations or recordings based on associated metadata. Search functionality is essential for efficient navigation and exploration of large datasets. It enables users to reference specific recordings or annotations quickly, improving the overall ease of use of the tool. Additionally, search capabilities enable users to filter recordings or annotations based on specific criteria, making it easier to identify unannotated files that should be included.

Annotation Exploration Whether the tool has annotation exploration capabilities. This means that the tool can display multiple annotations in a way that allows the user to visualise and compare several annotations simultaneously. In particular, I am interested in the ability to visualise annotations stemming from different recordings in the same workspace.

Flexible Spectrogram Whether the tool has a flexible spectrogram generation system. This means that the tool can generate spectrograms with different parameters, such as the window size, the window type, the overlap, the colour scale, etc.

Flexible Annotation Whether the tool has a flexible annotation system. This means that the tool can generate annotations of different types, such as point annotations, interval annotations, and bounding box annotations. Also, I require the ability to define custom tags, not restricted to species names or taxonomic terms.

Quality Assurance Whether the tool includes integrated tools to help with quality control. These are any tools that help the user to check the quality of the annotations and flag potential errors.

Training tools Whether the tool includes interactive components designed to assist in the training of novice annotators tailored to the current annotation objectives. Such components may include features that enable easy comparison of sounds to identify similarities and differences, or mechanisms to test the aural identification skills of an annotator. Providing training tools can be especially useful for inexperienced annotators, allowing them to develop and refine their skills more quickly.

Prediction Evaluation Whether the tool provides a mechanism for evaluating predictions against a set of ground truth annotations. Ground truth evaluation is essential for assessing the accuracy and reliability of automated annotation algorithms. By comparing the results of automated annotation against a known ground truth, it is possible to identify areas where improvements are needed.

Export Annotations Whether the tool allows exporting the annotations into a shareable format with a clear schema. This is important for researchers who need to use the annotations in other software or for training Deep Learning (DL) models.

Integrated Detectors This criterion evaluates whether the tool integrates automated detector capabilities. This means that the tool can use ML or otherwise to automatically generate annotations.

The evaluation of each tool was conducted by reading the documentation and

Table A.1: Evaluation of current annotation tools against established criteria. The total number and percentage of evaluated annotation tools that meet each of the established criteria.

	Total	Percentage
Open Source	28	62.2%
Self Hosted	41	91.1%
Collaborative Use	13	28.9%
Handling Large Datasets	23	51.1%
Rich Metadata Display	11	24.4%
Search Capabilities	9	20.0%
Annotation Exploration	7	15.6%
Flexible Spectrogram	26	57.8%
Flexible Annotation	4	8.9%
Quality Control	5	11.1%
Annotator Training	0	0.0%
Prediction Evaluation	1	2.2%
Integrated Detectors	14	31.1%

user guides provided by the tool developers, or by using the tool itself when possible. I acknowledge that this evaluation is not entirely objective and that the results may be biased by the experience of the authors.

Out of the 45 tools evaluated, none met all the established criteria. Notably, no tool included a component specifically designed to assist with annotator training, with the possible exception of tools that provided annotation instructions, such as Simpson et al., 2014. Only a small proportion of tools (less than 16%) included features for quality control, annotation exploration, and prediction evaluation (Table A.1). These findings suggest that the majority of previously developed audio annotation tools did not prioritise the creation of annotated datasets suitable for DL development. To see the full list of tools evaluated consult the `annotation_tool_comparison.csv` file in the supplementary material.

A.2 Annotation software design

`whombat` was designed with usability, scalability, and extensibility as priorities. Here I outline the key design decisions I made and the rationale behind them.

I believe that open source software fosters collaboration, innovation, and trans-

parency. Therefore, I decided to release our audio annotation tool as an open source project on a public repository. This allows other researchers, developers, and users to access, use, modify, and contribute to our codebase. I also provide documentation, examples, and tutorials to facilitate the adoption of our tool by the community.

I opted for a server-client configuration for our audio annotation tool as it affords the flexibility to host both backend and frontend on either separate machines or one machine, depending on resource availability and utilisation. This approach also facilitates exploiting web-based interface advantages such as portability and accessibility. In particular, I created the backend with a RESTful API that manages communications between client-server requests/responses while executing audio processing pipelines and saving outputs in a database. On the other hand, the frontend is responsible for displaying meaningful data to users as well as handling their interactions with it.

I chose `Python` (Van Rossum & Drake Jr, 1995) as the main language for the backend of our audio annotation tool because of its rich ecosystem of packages for scientific computing, data analysis, and web development. Our preference towards utilising `Python` also enables seamless integration with multiple Deep Learning (DL) tools and pipelines available for `Python`. Furthermore, I observed that using `Python` facilitates code sharing and collaboration due to its ease of learning and readability. `Python` has a large and active community of developers, researchers, and enthusiasts, which can provide support and feedback.

To implement our RESTful API, I employed the `FastAPI` (Ramirez, 2024) framework for its lightweight and flexible characteristics. For audio processing tasks, I utilised the `scipy` (Virtanen et al., 2020) and `numpy` (Harris et al., 2020) packages, which provide a wide array of functions for scientific computing and data analysis. All data produced and used by our audio annotation tool is saved in a relational database. `SQLite` was our default choice of database management system due to its lightness and efficiency in managing small to medium-sized datasets. Nonetheless, I acknowledge that certain users may require other database systems, such as `MySQL` or

PostgreSQL, contingent on their specific needs and constraints. Thus, I offer a configuration option to enable switching to a different database backend according to the preference of the user. Communication between the database and the backend was facilitated by the SQLAlchemy (Mike Bayer, 2023) package, providing a high-level interface for managing database systems. Moreover, I provide a Python API that enables direct interaction with the stored data, allowing users to create customised analysis pipelines or integrate data into DL pipelines. This feature provides flexibility and extensibility beyond the default functionality.

In selecting a language for the user-facing components of our audio annotation tool, I opted for TypeScript (Bierman et al., 2014), a superset of JavaScript that includes optional static typing to enhance code quality. For constructing the interface itself, I turned to React (Walke, 2023), a widely used and effective library that employs a declarative and component-based approach to building interfaces. This approach affords us greater consistency in design and allows us to reuse UI elements.

I wrote the audio annotation tool with the aim of making it easy to understand and extend. To that end, I added comprehensive documentation in all the main modules and functions, including detailed explanations of the inputs, outputs, and behaviour of each component (available at). I also provided examples of how to use the tool in practice, as well as clear instructions for setting up and configuring the tool. Additionally, I implemented unit and integration tests in the most critical parts of the software, to ensure correct behaviour and facilitate future development. These tests cover a wide range of scenarios and edge cases, and are automatically run whenever changes are made to the codebase. By providing clear documentation and robust testing, I hope to make it easier for users to understand and extend our tool, as well as contribute to the broader bioacoustics community.

Appendix B

Appendix for Chapter 3

B.1 Data split

To assess model performance in novel geographic locations, I used a location-based split strategy, assigning recordings to the development or test datasets based on their recording location. Having no overlapping recording sites between the development and test dataset means that background environments and recorded individuals are all different. Recordings that lacked location data or originated outside of Mexico were automatically assigned to the development set. All other recording locations were split using multi-label stratified splitting (Sechidis et al., 2011). This method treats the list of recorded species at each location as a set of labels and assigns locations to the development (75%) and testing sets (25%) while maintaining proportional species distribution. To ensure the test set reflects the acoustic diversity across Mexico, I generated multiple multi-label stratified split proposals and selected the one that maximised both the number of covered ecoregions (INEGI CONABIO, 2008) and the number of species in the test set (Figure B.1). This strategy aimed to capture a wide range of acoustic variation and species representation in the test set, providing a robust evaluation of the models' generalisation ability.

To ensure accurate and up-to-date taxonomic information, I standardised species names using the GBIF Backbone Taxonomy (Secretariat, 2023), updating any outdated names to the currently accepted versions. After splitting the data into test

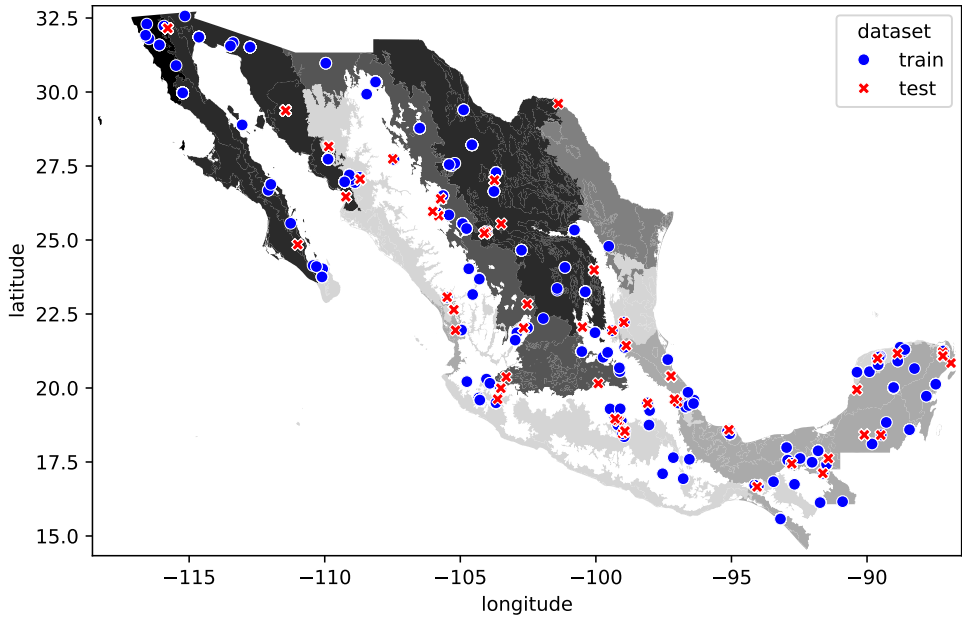


Figure B.1: Location-based dataset split. This map displays the distribution of recording sites across Mexico. Training sites are marked in blue circles, and testing sites are in red crosses. The underlying regions represent the ecoregions defined by INEGI CONABIO, 2008.

and development sets, some species occurred exclusively in one set or the other. The final test dataset contained 69 distinct species, while the development dataset contained 97. To conduct experiments with varying dataset sizes (ranging from 5 to 25 recordings per species) and ensure sufficient data for validation and testing, I selected a subset of 17 species. These species were selected based on having at least 30 recordings in the development set and at least 5 recordings in the test set. Table B.1 provides a detailed breakdown of the number of recordings per species in both the development and test sets.

Table B.1: Summary of the dataset used for bat call classification. For each species, the 'Recordings' columns show the total number of unique recordings and their breakdown into development and test datasets. The 'Calls' columns provide the total number of individual echolocation pulses annotated and their distribution across the development and test datasets.

	Recordings			Calls		
Species	Total	Development	Test	Total	Development	Test
Antrozous pallidus	192	158	34	3556	2920	636
Myotis velifer	150	113	37	2611	1938	673
Eptesicus fuscus	146	76	70	2615	1730	885
Balantiopteryx plicata	127	117	10	2567	2379	188
Artibeus jamaicensis	88	65	23	1176	710	466
Aeorestes cinereus	77	32	45	1266	698	568
Macrotus californicus	68	36	32	1621	1137	484
Leptonycteris yerbabuenae	63	34	29	2491	1560	931
Saccopteryx bilineata	53	39	14	1057	768	289
Pteronotus parnellii	51	34	17	1293	981	312
Molossus rufus	50	44	6	1005	869	136
Myotis californicus	49	39	10	884	736	148
Myotis yumanensis	43	35	8	1001	878	123
Tadarida brasiliensis	41	31	10	657	498	159
Corynorhinus townsendii	39	30	9	933	788	145
Natalus mexicanus	36	31	5	460	422	38
Peropteryx macrotis	36	31	5	608	500	108

B.2 Model architecture

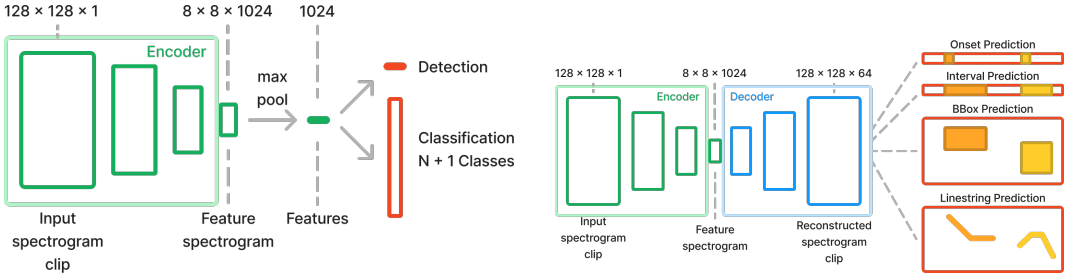
This appendix details the architecture of the Convolutional Neural Network (CNN) models used for bat call detection and classification. I first describe the base CNN model, which performs detection and classification. Then, I detail the decoder component incorporated into the model to enable the localization of bat calls within spectrograms.

All model variants use the same 10-layer CNN encoder to extract features from the input spectrogram. This encoder computes a 1024-dimensional feature vector used for both detection and classification. The encoder architecture consists of four blocks, each comprising two convolutional layers with ReLU activation functions, followed by a 2x2 max-pooling layer. Batch normalization is applied after each ReLU activation for improved training stability (Ioffe & Szegedy, 2015). The number of filters in each convolutional layer increases progressively: 64, 64, 128, 128, 256, 256, 512, 512, 1024 and 1024. I chose this model architecture due to its simplicity

and common use for acoustic classification (Mac Aodha et al., 2018).

The outputs of the encoder are then used for detection and classification. The encoder takes a $128 \times 128 \times 1$ spectrogram clip, $\mathbf{S}$, as input and produces a $8 \times 8 \times 1024$ representation, which I refer to as the feature spectrogram. A max-pooling operation is applied to the feature spectrogram to obtain a 1024-dimensional feature vector, $\mathbf{f}$. This feature vector is then passed to two separate heads for detection and classification (Figure B.2a). These heads consist of fully connected layers with 1 and $n+1$ output neurons, respectively, where n is the number of target species. The outputs of these heads are a detection value, $\widehat{b}$, representing the confidence score that $\mathbf{S}$ contains a bat call, and a classification vector, $\widehat{\mathbf{c}}$, whose elements represent the confidence scores for each species being present in $\mathbf{S}$. More precisely, $\sigma(\widehat{b}) = \mathbb{P}(bat \mid \mathbf{S})$, $\phi(\widehat{c})_0 = \mathbb{P}(\text{no bat or other species} \mid \mathbf{S})$ and $\phi(\widehat{c})_i = \mathbb{P}(\text{species}_i \mid \mathbf{S})$, where σ and ϕ are the sigmoid and softmax functions, respectively. This scheme allows the network to detect bat calls from non-target species when $\sigma(\mathbf{b}) = 1$ and $\phi(\widehat{\mathbf{c}})_0 = 1$. While this encoding of unknown classes is relatively simple, exploring more sophisticated open-set recognition techniques is left for future work. However, it is important to note that deep learning model confidence scores are often poorly calibrated, meaning they do not accurately reflect the true probability of a correct prediction (Guo et al., 2017; Dussert et al., 2024). Therefore, further calibration techniques may be needed to improve the reliability of these scores (Wood & Kahl, 2024).

To enable the prediction of sound event locations within the clip, I incorporated a CNN decoder component. This component uses the feature spectrogram to predict a mask with the same dimensions as the input spectrogram, indicating which pixels belong to a target sound event. The decoder consists of four blocks, each with two convolutional layers with ReLU activation functions, followed by a transposed convolutional layer. Batch normalization is applied after each ReLU activation. The transposed convolutional layers progressively upsample the feature spectrogram to the original input spectrogram size. The number of filters in each convolutional layer decreases progressively: 128, 128, 64, 64, 32, 32, 16, and 16. This configuration



(a) Detector and classifier architecture. **(b)** Reconstruction of the location labels.

Figure B.2: Overview of the detector and classifier model architecture. Firstly, the input spectrogram is encoded into a $8 \times 8 \times 1024$ tensor. Then a 1024 feature vector is computed by applying a max pool operation. This feature vector is used to predict the presence and species class of calls within the input spectrogram. A spectrogram clip is fed into the encoder and converted into a feature spectrogram with reduced dimensions. The feature spectrogram is then upsampled to the original size by the decoder network. The reconstructed spectrogram is then used to predict the location labels associated to the detailed annotations.

was determined through a hyperparameter sweep conducted after fixing the encoder architecture, using classification performance on the validation set as the selection criterion. Finally, a 1x1 convolutional layer combines the 16 output channels into a single channel representing the logits of the probability score that each pixel contains a bat call.

B.3 Model training

To train the models, I used a combined loss function incorporating both detection and classification objectives. The detection loss was calculated as the binary cross-entropy between the model's prediction of bat presence in the input spectrogram and the ground truth label. The classification loss was calculated as the multi-class cross-entropy between the model's confidence scores for each species and the corresponding ground truth label. For all variants the annotations were used to determine the presence and species of a bat call in the input spectrogram. The training loss for the recording and clip variants was the sum of the detection and classification losses. However, the classification loss was set to 0 for examples where no bat was present in the input spectrogram.

To incorporate the localisation task, I added a localisation loss component to the overall loss function. I used the focal loss between the predicted mask generated by

the decoder and the binary masks derived from the annotations. Focal loss is well-suited for dense prediction tasks like this, as it addresses potential class imbalance issues in the mask (Lin et al., 2017). I used the default focal loss parameters, as preliminary experiments showed no significant improvement with alternative parameter values. The binary masks used as training targets differed depending on the model variant. These were generated from the annotations, potentially after converting the line-string annotations to a simpler type, as described in the Methods section. The total loss for the `onset`, `onset-offset`, `bounding-box`, and `line-string` models was the sum of the detection, classification, and localisation losses.

To mitigate overfitting, particularly crucial with small training datasets, I employed data augmentation techniques commonly used in bioacoustic tasks (Lauha et al., 2022; Kahl et al., 2021; Park et al., 2019). These techniques included time and frequency masking, Gaussian noise addition, artificial echo addition, and random image cropping. Time and frequency masking involve randomly selecting a band of pixels in the time or frequency direction of the input spectrogram and setting their values to the average spectrogram value. I used random bands with a maximum width of 10 pixels out of the total 128 pixels. Gaussian noise addition involve adding random noise to each spectrogram pixel drawn from a Gaussian distribution with a signal-to-noise ratio of 3. Artificial echoes were added by overlaying a time-shifted and attenuated version of the original spectrogram. Finally, random image cropping involve randomly selecting a portion of the spectrogram and rescaling it to the original size. This process could result in stretching of the time and frequency axes, but the size of the selected crop was limited to a minimum of 90% of the original width and height to minimise distortion. While bat echolocation calls typically exhibit frequency specificity, some degree of frequency plasticity is observed (Montauban et al., 2021). Consequently, the frequency shifts introduced by random cropping help emulate natural variations within the call's frequency range. However, as with any data augmentation method, this approach may not fully represent the complex, coordinated adjustments in frequency and duration that occur naturally.

Appendix C

Appendix for Chapter 4

C.1 Model architecture details

This section provides additional details on the BatDetect2 model architecture, expanding upon the description in the main text. The model employs a 3-layer U-Net-style architecture (Ronneberger et al., 2015), incorporating an encoder, a decoder, and skip connections between them. A self-attention layer (Vaswani, 2017), denoted as `self_attn`, is incorporated in the central bottleneck of the model, enabling it to leverage information across extended timescales. This self-attention layer utilises a feature dimension of 256 and does not employ positional encoding. The model incorporates two specialised building blocks: `CoordConvDown` and `CoordConvUp`. The `CoordConvDown` layer performs the following sequence of operations: appending frequency coordinate information, 2D convolution, 2×2 max-pooling for downsampling, batch normalisation (BN) (Ioffe & Szegedy, 2015), followed by a ReLU non-linearity (Nair & Hinton, 2010). The `CoordConvUp` layer performs a similar, but inverse, set of operations, effectively upsampling the input tensor. This involves 2D bilinear upsampling, appending frequency coordinates, 2D convolution, batch normalisation, followed by a ReLU activation. The complete architecture of the BatDetect2 model is detailed in Table C.1.

Following the model output, a non-maximal suppression operation is applied, implemented as two-dimensional max-pooling with a 9×9 kernel. The model then

reports the top 200 events for each one-second segment of input audio, ranked by detection probability. Although the model can process input sequences of arbitrary length, in practice, it is recommended to segment longer audio files into clips of less than two seconds for independent processing.

After output, I run a simple non-maximal suppression which is implemented as a two dimensional max pooling operation with a kernel size of 9×9. The model then reports the top 200 events, ordered by detection probability, for each one second of input audio. While the model can operate on arbitrary length sequences, in practice it is best to chunk longer input audio files into clips that are less than two second long, and then process each clip independently.

Table C.1: Description of the full architecture for BatDetect2 model. The values for input and output size refer to the feature dimension, height, and width of the respective tensors (e.g. (1, 128, 512) is one feature channel, with height 128 and width 512). The kernel size is represented as height and width. In the case where two tensors are added together for the input to a layer, this is simply performed using an element wise addition. The model outputs a $C+1$ dimensional vector for each location in time and frequency, where $C+1$ represents the number of classes plus one additional class for background, i.e. 'Not bat'. The model also outputs an additional two dimensional vector for each location which encodes the predicted width (i.e. duration) and height (i.e. frequency range) of any echolocation event at that location in time and frequency.

layer name	input	layer type	input size	output size	kernel size
Encoder					
conv_down_0	spectrogram	CoordConvDown	(1, 128, 512)	(32, 64, 256)	(3,3)
conv_down_1	conv_down_0	CoordConvDown	(32, 64, 256)	(64, 32, 128)	(3,3)
conv_down_2	conv_down_1	CoordConvDown	(64, 32, 128)	(128, 16, 64)	(3,3)
Bottleneck					
conv_3	conv_down_2	Conv2d, BN, ReLU	(128, 16, 64)	(256, 16, 64)	(3,3)
conv_1d	conv_3	Conv2d, BN, ReLU	(256, 16, 64)	(256, 1, 64)	(16,1)
self_attn	conv_1d	Self-Attention	(256, 1, 64)	(256, 1, 64)	n/a
repeat_vert	self_attn	Repeat Vertical	(256, 1, 64)	(256, 16, 64)	n/a
Decoder					
conv_up_0	repeat_vert + conv_3	CoordConvUp	(256, 16, 64)	(64, 32, 128)	(2,2)
conv_up_1	conv_up_0 + conv_down_1	CoordConvUp	(64, 32, 128)	(32, 64, 256)	(2,2)
conv_up_2	conv_up_1 + conv_down_0	CoordConvUp	(32, 64, 256)	(32, 128, 512)	(2,2)
Output					
conv_op_0	conv_up_2	Conv2d, BN, ReLU	(32, 128, 512)	(32, 128, 512)	(3,3)
pred_class - $\hat{Y}$	conv_op_0	Conv2d, Softmax	(32, 128, 512)	($C+1$, 128, 512)	(1,1)
pred_size - $\hat{S}$	conv_op_0	Conv2d, ReLU	(32, 128, 512)	(2, 128, 512)	(1,1)

C.2 Training loss details

In this section I describe the training loss used by BatDetect2. The loss function is composed of three main terms and is inspired by those used in the CenterNet method for object detection in images (Zhou et al., 2019). The combined losses encourage the model to correctly predict the location, in frequency and time, of each echolocation call, the duration and frequency range of the call, and the species that is responsible for making the call.

Let us denote $\mathbf{x} \in R^{H \times W}$ as the input spectrogram, with height H and width W. Here, height refers to the number of frequency bins and width is the number of temporal bins in the spectrogram. Prior to the final post-processing step (i.e. non-maximal suppression), the model outputs two tensors, $\hat{Y} \in [0,1]^{H \times W \times C+1}$ and $\hat{S} \in \mathbb{R}_{\geq 0}^{H \times W \times 2}$. Here, C is the total number of species of interest, while the additional class is used to represent the background class (i.e. no bat present). $\hat{Y}$ is the predicted species class probabilities and $\hat{S}$ contains the predicted size of any echolocation call estimated to be present. At training time the model has access to the ground truth values for Y and S. Both $\hat{Y}$ and $\hat{S}$ contain an estimated value for each location in time and frequency space in the input spectrogram. For example, for a given frequency band f and time step t, $\hat{S}_{ft1}$ encodes the predicted duration of the call (i.e. $t_{\text{end}} - t_{\text{start}}$), and $\hat{S}_{ft2}$ encodes the predicted frequency range of the call (i.e. $f_{\text{max}} - f_{\text{min}}$). For a description of how Y and S are generated, please see the main text.

Additionally, let us define $\hat{E}_{ft} = \sum_{c=1}^{C} \hat{Y}_{ftc}$, and similarly $E_{ft} = \sum_{c=1}^{C} Y_{ftc}$. $\hat{E}$ and E represent predicted and ground truth class-agnostic echolocation call scores, i.e. 'Bat' versus 'Not bat'. Note that for $\hat{E}$ and E, the sum over does not include the background class. These additional terms are included as there are many instances in which the annotators have difficulty determining the correct species for a given call, and thus they can only label the event with the generic 'Bat' class label. This supervision can still be leveraged by allowing the model to determine which species may be present.

The goal during training is to minimise the difference between the estimated

$\hat{E}$, $\hat{Y}$, and $\hat{S}$ and the respective ground truth values E, Y, and S. If successful, the model will be able to correctly predict the location in time and frequency of any echolocation call along with the species of the bat that generated the call.

C.2.0.1 Losses

The first loss encourages the model to correctly discriminate between bat echolocation calls and non-bat calls, i.e. background noise or other vocalising species. To achieve this, I use the focal loss (Lin et al., 2017), specifically, the keypoint variant of the focal loss from Law & Deng (2018), which is defined as:

$$L_{det} = -\frac{1}{N}\sum_{f=1}^{H}\sum_{t=1}^{W}\begin{cases}(1-\hat{E}_{ft})^{\alpha}\log(\hat{E}_{ft}) & \text{if } E_{ft}=1\\ (1-E_{ft})^{\beta}(\hat{E}_{ft})^{\alpha}log(1-\hat{E}_{ft}) & \text{otherwise,}\end{cases} \tag{C.1}$$

where N is the number of echolocation events in the spectrogram.

The next loss penalises the model for assigning the wrong species label to a detected echolocation call. This loss is similar L_{det}, but instead of only discriminating between 'Bat' and 'Not bat', this loss encourages the model to predict the correct species label for each echolocation call. I use a masked version of the loss which is only applied to locations in the spectrogram where there is a echolocation call present, i.e. where $E_{ft} > 0$.

$$L_{class} = -\frac{1}{N}\sum_{f=1}^{H}\sum_{t=1}^{W}\sum_{c=1}^{C+1}\begin{cases}0 & \text{if } E_{ft}=0\\ (1-\hat{Y}_{ftc})^{\alpha}\log(\hat{Y}_{ftc}) & \text{if } E_{ft}>0 \text{ and } Y_{ftc}=1\\ (1-Y_{ftc})^{\beta}(\hat{Y}_{ftc})^{\alpha}log(1-\hat{Y}_{ftc}) & \text{otherwise.}\end{cases} \tag{C.2}$$

The final component of the loss penalises the model for incorrectly predicting the 'size' of the predicted bounding box which overlaps with a ground truth echolocation

call. Like L_{class}, this loss is only applied to locations in time and frequency where an echolocation call in the training set has been annotated.

$$L_{size} = \frac{1}{N} \sum_{f=1}^{H} \sum_{t=1}^{W} \begin{cases} \left|\hat{S}_{ft1} - S_{ft1}\right| + \left|\hat{S}_{ft2} - S_{ft2}\right| & \text{if } \sum_k S_{ftk} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{C.3}$$

Here, $\sum_k S_{ftk} > 0$ simply indicates that this size loss is only applied where there is a echolocation call present.

The final combined loss to minimise during training is

$$L = \lambda_1 L_{det} + \lambda_2 L_{class} + \lambda_3 L_{size}. \tag{C.4}$$

The loss is summed over each spectrogram in a given input training batch. During training, I set λ_1, λ_2, λ_3 to 1.0, 2.0, and 0.1 respectively, and for both focal losses I set $\alpha = 2$ and $\beta = 4$.

C.3 Audio datasets

Here I provide additional details of the different datasets used in for training and evaluation.

C.3.1 UK data

In total there are 17 species in the UK dataset. This is the total number of species which are known to be breeding in the UK. The data comes from 2,809 audio files, and contains a total of 34,635 annotated echolocation calls. The data has been collected using a variety of devices and was provided by a number of different sources. There are six main sources of data, where each source constitutes a single organisation or individual that provided multiple different audio files. This diversity is important as it maximises the variation in the training set, with the ultimate aim of having better generalisation performance at test time. The majority of the recordings were made in the UK, but there were also some additional files included from the species of interest that were recorded elsewhere (e.g. Europe). The annotation process prioritised annotating only one clip, at most two seconds in duration, from each original input recording, rather than densely annotating long, multi-second audio files. This was also performed in order to increase the data diversity, as there can often be a large amount of self-similarity within the same longer recording. As a result, the clipped files vary in duration from between 0.4 to two seconds, and the average duration is just over one second.

In order to increase robustness to background noise, I also supplement the UK species audio by including additional recordings that are either empty (i.e. did not contain bats) or where a bat was present but of unknown species. The empty recordings were collected in London, UK, using the custom built IoT smart sensor from Gallacher et al., 2021. In total there are 345 three second files in this set. The second set of extra data came from the iBats Program (Jones et al., 2013) as was adapted from Mac Aodha et al., 2018. This set includes 4,225 files of 0.384 seconds in duration and contains 6,842 annotated bat calls. This data was recorded using Tranquility Transect detector using a time expansion factor of ten.

With the exception of the background and bat-only recordings, the rest of the files were recorded to contain confirmed species at the file level. Experienced annotators, familiar with the characteristics of UK bat echolocation calls, drew bounding boxes around each individual echolocation call and assigned species labels, where possible, based on the file level confirm species. When unsure of the species label, they annotated the call using the generic 'Bat' class label.

The BatDetect2 model predicts the location of the lower left corner for each echolocation call in an input recording. For the two constant call frequency-based species in the UK, *Rhinolophus ferrumequinum* and *Rhinolophus hipposideros*, there was a high degree of variability in the position of the lower left corner of the call. This happens as a direct result of the recording quality, characteristics of the local environment, and the distance of the bat from the microphone As a result, it was often difficult to determine the exact lower frequency for these two species. To overcome this issue, I standardised the lower and upper frequency for each of the these species by setting them to per-species mean values, where the means were computed on the training sets.

I constructed two splits for the UK dataset. Both splits contain the same number of calls overall, and only differ in how the data is distributed between their respective training and test sets. As noted earlier, there are six main sources of data for the UK bat recordings. The first split, referred to as UK_{same}, simply shuffles the files randomly into training and test sets and ensures that there is a maximum of four recordings (i.e. files not calls) per species, per data source, in the test set. This results in a split with 7,010 training files and 369 test files (Table C.2).

The second split, UK_{diff}, is more challenging. Here I simulate a difficult real world setting where an entire data source is held out for validation. I remove one of the largest sources, which leaves 5,911 training files and 1,468 test files (Table C.3). This increases the difficulty due to the reduction in the training set size as well as increasing any potential domain gap that may exists between the train and test sets. This test set does not contain one of the species, *Pipistrellus nathusii*, as it was not

Table C.2: Number of annotated echolocation calls in the UK dataset using the UK_{same} split. There are a total of 7,010 and 369 training and test files, each containing 36,955 and 4,522 annotated echolocation calls respectively.

id	species name	num train calls	num test calls
0	Bat	8112	203
1	*Barbastellus barbastellus*	864	179
2	*Eptesicus serotinus*	2374	211
3	*Myotis alcathoe*	695	183
4	*Myotis bechsteinii*	648	222
5	*Myotis brandtii*	1775	166
6	*Myotis daubentonii*	5729	640
7	*Myotis mystacinus*	2430	384
8	*Myotis nattereri*	2384	328
9	*Nyctalus leisleri*	1056	85
10	*Nyctalus noctula*	310	99
11	*Pipistrellus nathusii*	1224	236
12	*Pipistrellus pipistrellus*	1653	245
13	*Pipistrellus pygmaeus*	2171	396
14	*Plecotus auritus*	917	193
15	*Plecotus austriacus*	690	177
16	*Rhinolophus ferrumequinum*	1915	290
17	*Rhinolophus hipposideros*	2008	285

possible to capture any recordings of it. Note that in both cases the data is still split at the file level (as opposed to individual call level). This minimises any potential overlap between the training and test sets.

Figures C.1 depicts a per-class average spectrogram for each species in the training set for the UK_{diff} split. Note that this averaging hides many of the recording specific difficulties and noise. It is thus is only provided for illustrative purposes as it shows the dominant 'shape' of the call for each species.

C.3.2 Yucatan data

This dataset consists of 285 passive recordings gathered in the Yucatan peninsula in Mexico as part of a field study conducted between 2004 and 2006 (MacSwiney G. et al., 2008). A Pettersson D980 bat detector device was used to detect and record bat calls. The device was active throughout three ten-minute periods at night, in a total of eight sites and covering twelve sampling nights per site. When active, and if a bat

Table C.3: Number of annotated echolocation calls in the UK dataset using the UK_{diff} split. There are a total of 5,911 and 1,468 training and test files, each containing 24,315 and 17,162 annotated echolocation calls respectively.

id	species name	num train calls	num test calls
0	Bat	7501	814
1	*Barbastellus barbastellus*	468	575
2	*Eptesicus serotinus*	403	2182
3	*Myotis alcathoe*	374	504
4	*Myotis bechsteinii*	241	629
5	*Myotis brandtii*	351	1590
6	*Myotis daubentonii*	3998	2371
7	*Myotis mystacinus*	1378	1436
8	*Myotis nattereri*	2610	102
9	*Nyctalus leisleri*	695	446
10	*Nyctalus noctula*	209	200
11	*Pipistrellus nathusii*	1460	0
12	*Pipistrellus pipistrellus*	868	1030
13	*Pipistrellus pygmaeus*	1461	1106
14	*Plecotus auritus*	528	582
15	*Plecotus austriacus*	331	536
16	*Rhinolophus ferrumequinum*	717	1488
17	*Rhinolophus hipposideros*	722	1571

call was detected, the device would record for three seconds and a time expanded version would be stored on a magnetic tape. The recordings were then cut into one second clips, resulting in a total of 1,193 audio files.

The species identification of the bat calls was made in two phases. For the original study, all recordings were reviewed manually. From each recording, at most five representative echolocation calls per detected species was selected and analyzed using Bat Sound Pro 3.10. The species of each call was then identified through comparison to a bat call library of captured bats from the same study. Please consult MacSwiney G. et al., 2008 to see the full details of their identification protocol.

In the second phase I annotated all missing bat calls using the annotation interface. Bounding boxes were drawn around each detected bat call in the spectrogram. Species identification was performed by comparing to the previously annotated calls. In order to gain confidence on the species labels for the additional boxes, I evaluated

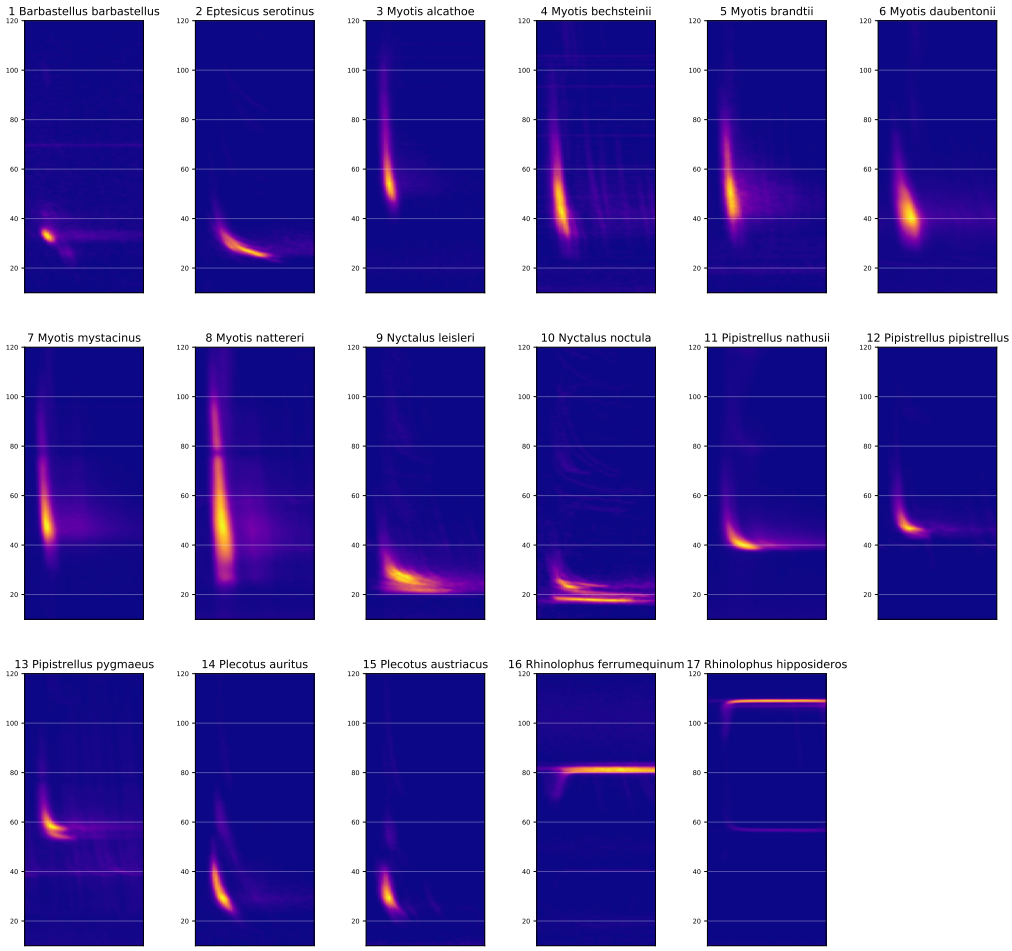


Figure C.1: Visualisation of the UK_{diff} species. Here, each sub-image represents the average spectrogram for each echolocation call from that species in the training set. The vertical axis represents kHz, and spans 10kHz to 120kHz, and the time duration for each spectrogram is 33.5 milliseconds.

my identification accuracy. A species label was added only if I could accurately identify said species (precision above 95%). In cases where it was not possible to determine the species, the call was labelled using the generic 'Bat' class. A recording was fully annotated when all bat echolocation calls were marked with a bounding box and all recognisable calls were tagged with its species, or the generic, label. This resulted in a total of 1,193 audio clips that were fully annotated and kept as part of the dataset. Three species (*Pteronotus personatus*, *Molossops greenhalli*, and *Molossus sinaloae*) were excluded as they only appeared in fewer than seven distinct recordings. The annotations for these species was set to the generic 'Bat' class. The final annotated dataset consists of 10,020 individual bat echolocation calls

with bounding box annotations from 17 different species.

To train and evaluate the detection and classification models I split the dataset into distinct training and testing subsets. To minimise any leakage from the test to the train set, I opted to split the data at the recording level, i.e. I avoided including one-second clips from the same recording in the training and testing subsets. The test set contains ∼20% (282 audio clips) of all recordings while the remaining ∼80% (911 audio clips) was used for training (Table C.4). In order to maintain the distribution of calls per species between the full dataset and the testing and training datasets, I labelled each recording with all its occurring species and used a stratified sampling method for multilabel datasets (Sechidis et al., 2011).

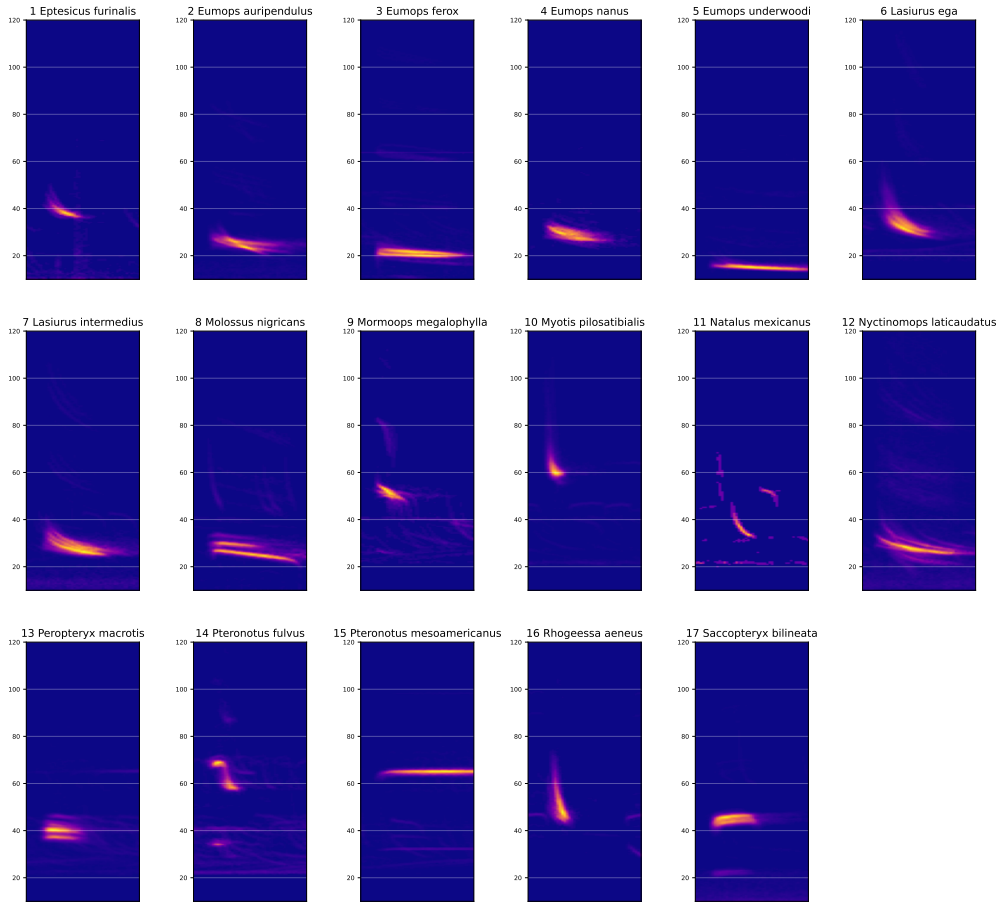


Figure C.2: Visualisation of Yucatan species. Here, each sub-image represents the average spectrogram for each echolocation call from that species in the training set. The vertical axis represents kHz, and spans 10kHz to 120kHz, and the time duration for each spectrogram is 33.5 milliseconds. Note that for some species we have limited numbers of example calls which results in noisy average spectrograms.

Table C.4: Number of annotated echolocation calls in the Yucatan dataset. In total there are 911 and 282 training and test files, which contain 7,677 and 2,343 individual calls respectively.

id	species name	num train calls	num test calls
0	Bat	3556	1236
1	*Eptesicus furinalis*	94	4
2	*Eumops auripendulus*	156	36
3	*Eumops ferox*	60	24
4	*Eumops nanus*	66	33
5	*Eumops underwoodi*	36	18
6	*Lasiurus ega*	250	69
7	*Lasiurus intermedius*	106	31
8	*Molossus nigricans*	65	25
9	*Mormoops megalophylla*	172	30
10	*Myotis pilosatibialis*	519	90
11	*Natalus mexicanus*	62	26
12	*Nyctinomops laticaudatus*	98	23
13	*Peropteryx macrotis*	1036	322
14	*Pteronotus fulvus*	509	167
15	*Pteronotus mesoamericanus*	345	81
16	*Rhogeessa aeneus*	166	36
17	*Saccopteryx bilineata*	381	92

C.3.3 Australia data

The Australian dataset used to train and test the model was taken from a bat call reference library collected by a bat expert. The subset used consists of a set of 14 bat species which have a sympatric distribution in the major cotton growing region on the north west plains of New South Wales and adjacent areas in central southern Queensland. Bat calls were recorded in the field from individuals released after capture, following positive species identification. A custom made digital ultrasound recorder from Nanobat Systems was used to record echolocation calls in 5 second sequences with a sampling rate 500 kHz and stored as 16 bit WAVs. Bats were recorded for as long as they flew around the release site until out of recording range. The resulting files were analysed and edited using Audacity 3.2.0 to find echolocation pulse sequences with good signal to noise ratio, undistorted waveforms and as close to search phase as possible. Edited wav files were then accumulated from the release

recordings of multiple individuals of the same species and across the species group.

These audio files had an average length of 3.29 seconds, with the shortest being 0.23 seconds and the longest being 10 seconds in duration. All annotated pulses were labelled by species since the original sequences were obtained from individually released bats, identified to species level. The only exception comes from the Ozimops species where the low release number of individuals (rarely caught) was augmented by identifying species from additional field recordings of bat activity at night. This was done manually by conventional sound analysis of field recordings taken from various study areas and using an experienced bat bioacoustics expert familiar with this genus. There were some instances where multiple species may have been present in a given file, and thus were potentially incorrectly attributed to the wrong species label.

The data was randomly split at the file level, with 80% of the recordings for a species staying the train set, and the rest in the test. This resulted in 220 training and 60 testing files (Table C.5).

Table C.5: Number of annotated echolocation calls in the Australia dataset. In total there are 220 and 60 training and test files, which contain 4,569 and 1,327 individual calls respectively.

id	species name	num train calls	num test calls
0	Bat	180	18
1	*Austronomus australis*	125	35
2	*Chalinolobus gouldii*	568	146
3	*Chalinolobus morio*	429	155
4	*Chalinolobus picatus*	327	157
5	*Nyctophilus corbeni*	537	101
6	*Nyctophilus geoffroyi*	179	41
7	*Nyctophilus gouldi*	363	97
8	*Ozimops petersi*	149	42
9	*Ozimops planiceps*	142	52
10	*Ozimops ridei*	122	64
11	*Saccolaimus flaviventris*	131	40
12	*Scotorepens balstoni*	232	120
13	*Scotorepens greyii*	273	90
14	*Vespadelus vulturnus*	812	169

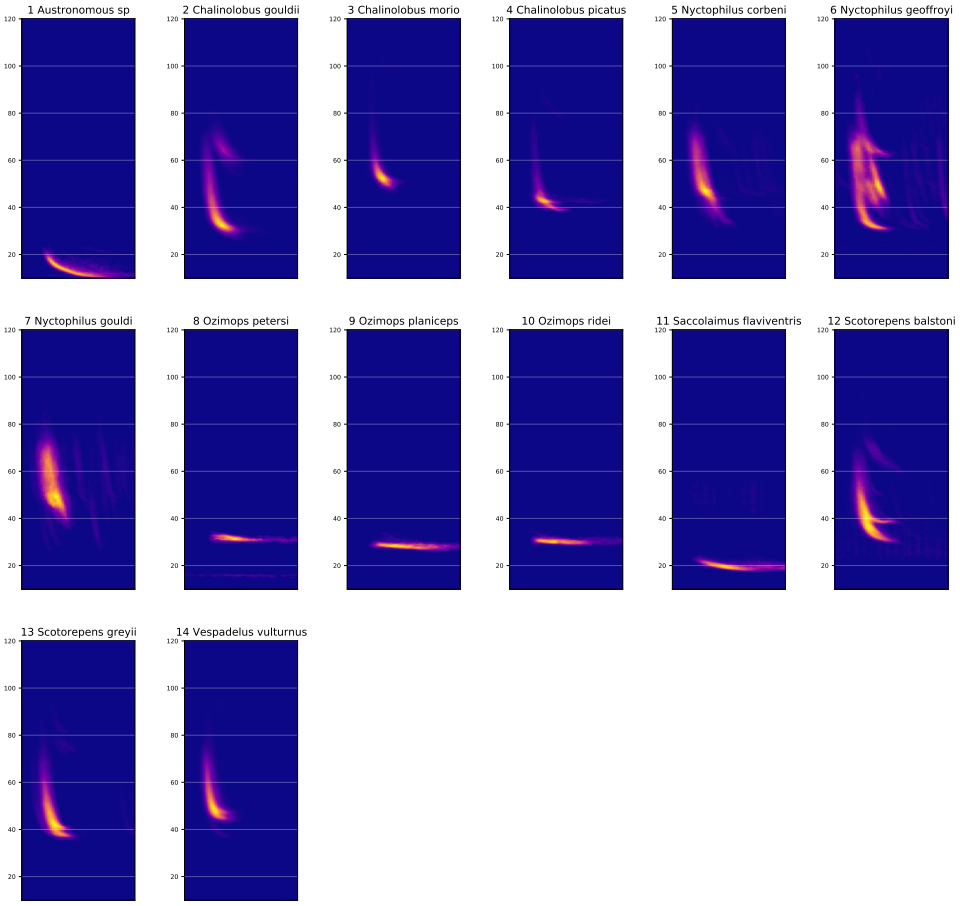


Figure C.3: Visualisation of Australian species. Here, each sub-image represents the average spectrogram for each echolocation call from that species in the training set. The vertical axis represents kHz, and spans 10kHz to 120kHz, and the time duration for each spectrogram is 33.5 milliseconds.

C.3.4 Brazil data

Data for this study was collected between January and March 2019 in south-eastern Brazil. The data used for training is a subset of acoustic data collected using AudioMoth (Hill et al., 2019) recorders which were set to record at a sampling rate of 395 kHz for one minute every five minutes between 22:00 and 04:00. The recorders were deployed on coffee plantations and in adjacent forest fragments. The final dataset consists of 320 ten second audio recordings.

As no species labels were available for this dataset, I opted to group the calls into groups based on their dominant frequency. Specifically, calls were initially labelled to genus level where quality allowed, but were later merged to a coarser call type

groups. This resulted in three distinct sonotypes, along with the generic bat class which served as an additional class for cases where it was not possible to identify calls to one of the previously mentioned three groups.

I randomly assigned ∼80% of the audio files (256 files) to the training set and the remaining ∼20% (64 files) to the test set. This resulted in a total of 7,989 and 2,010 calls in the respective sets (Table C.6).

Table C.6: Number of annotated echolocation calls in the Brazil dataset. In total there are 256 files in the training set and 64 in the test set. In both cases the files are ten seconds in duration.

id	species name	num train calls	num test calls
0	Bat	1646	619
1	Group One	2168	490
2	Group Two	2993	742
3	Group Three	1182	159

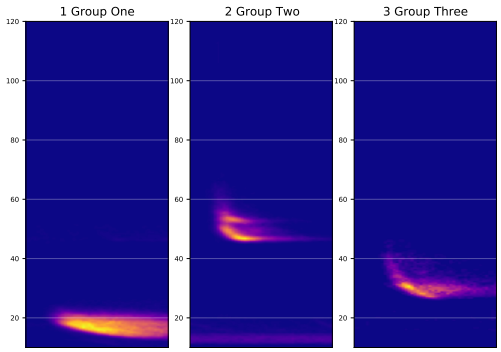


Figure C.4: Visualisation of the Brazil data. Here, the spectrograms do not represent species, but instead three distinct groups of calls. Each sub-image represents the average spectrogram for each echolocation call from that group in the training set. The vertical axis represents kHz, and spans 10kHz to 120kHz, and the time duration for each spectrogram is 33.5 milliseconds.

C.4 Full performance report

In this section, I present a comprehensive report of the BatDetect2 performance across all five datasets (Figure C.5). For each dataset, precision-recall curves are presented for each species. Additionally, per-genus precision-recall curves are displayed. These curves are generated by summing the predicted probabilities of all species within a genus to obtain a genus-level probability. Finally, file-level confusion matrices are presented. These matrices are generated by assigning each call within a file to the species with the highest predicted probability and then comparing these assignments to the ground truth species labels at the file level. I exclude files containing multiple species from the confusion matrix computation.

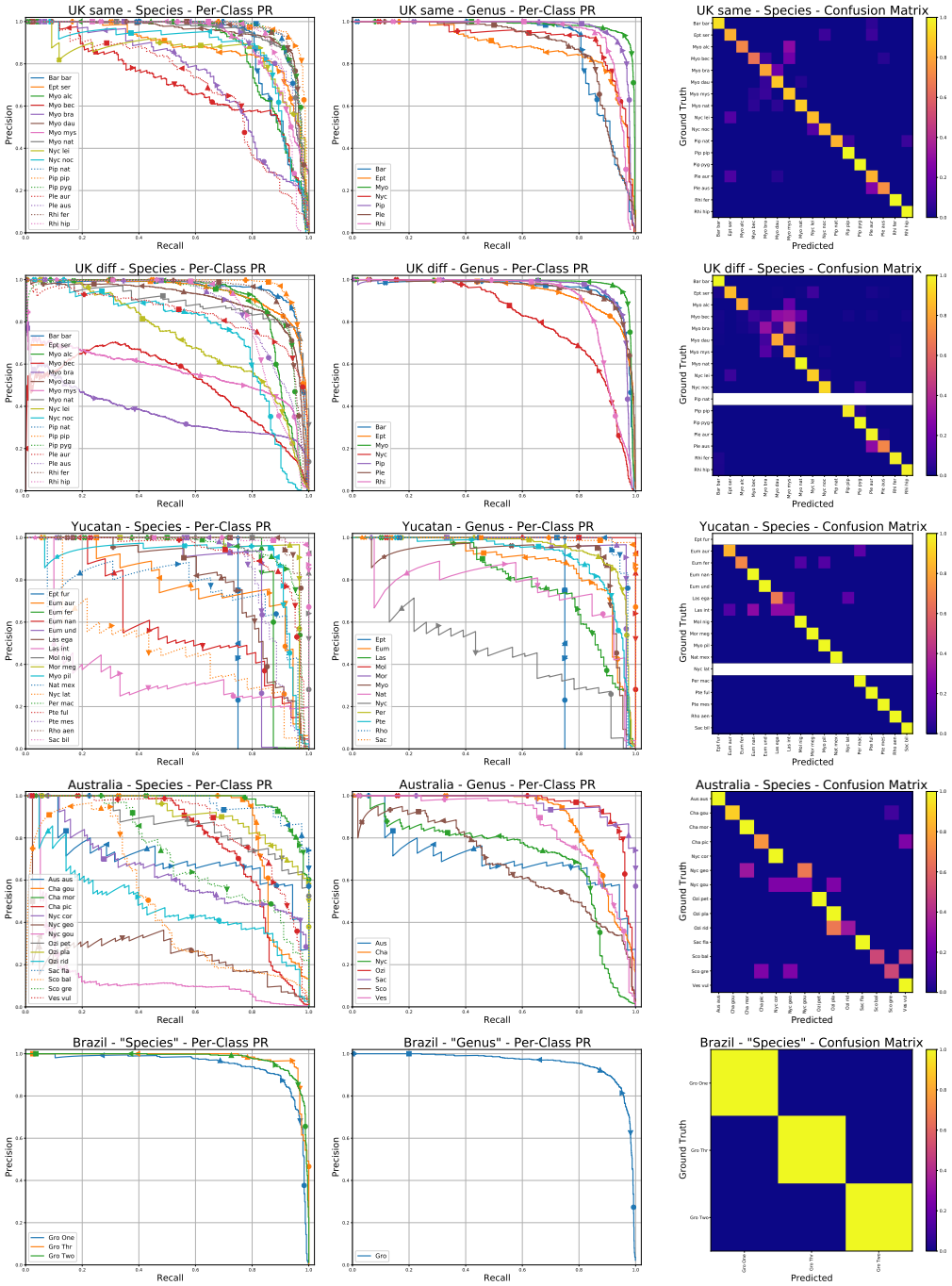


Figure C.5: Precision-recall (PR) and confusion matrices for our BatDetect2 model for the five different test sets. The first column depicts the per-species precision-recall curves and the second column is the per-genus equivalent. The third column illustrates the file-level confusion matrix, where white rows indicate that there were no species of that type in the filtered test set. Each row depicts a different dataset.

C.5 Self attention mechanism

This section provides an illustrative example of the self-attention mechanism employed within the BatDetect2 model. During the processing of an input spectrogram, the self-attention mechanism enables the model to identify and weight the most relevant time steps for predicting the species present at a given time point. The attention module operates solely along the temporal dimension. Figure C.6 illustrates the self-attention mechanism in action for a sample audio file. At each time point, the module computes self-attention scores against all other time steps in the input sequence. This process allows the model to leverage global contextual information across the entire input sequence when estimating the species present at a specific time point.

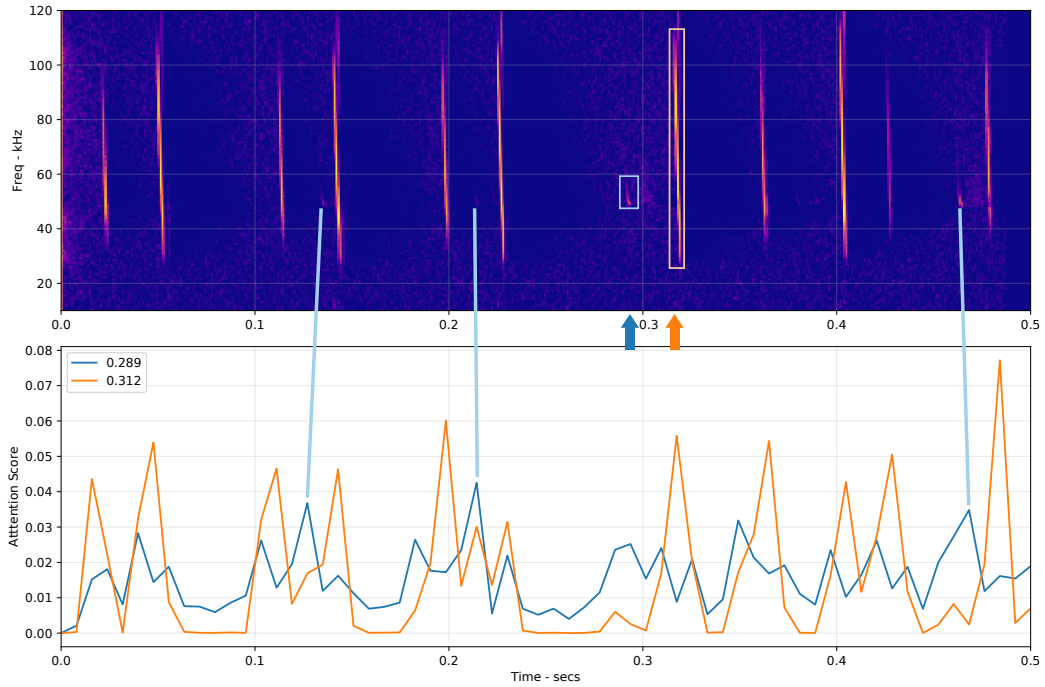


Figure C.6: Visualisation of the self-attention scores for one audio file from the UK dataset. Here we show the attention weights for only two locations in the input - at 0.289 and 0.312 seconds in the input spectrogram. We denote these two time points with a blue and orange arrow respectively, along with showing bounding boxes on the calls. The attention scores corresponding to the two time points are illustrated with blue and orange lines on the bottom plot. The orange line shows high attention for the other *Myotis* calls and the blue line indicates that the model places more attention on the other, less prominent, *Pipistrelle* calls. Note, there appears to be a very faint *Pipistrelle* call before the 0.4 second time step that the model has a low attention score for.

Appendix D

Appendix for Chapter 5

D.1 Deployment configurations

The `acoupi` framework offers flexible configuration options, enabling customisation of both *acoupi_batdetect2* and *acoupi_birdnet* deployments. Key configurable parameters include: (1) Microphone: selection of the recording device and its sampling rate (e.g., 44.1kHz for birds, 250kHz for bats). (2) Recording: definition of the duration of each contiguous audio recording triggered by the system, with a configurable schedule defining the start and end times during which recordings are permitted. (3) Task intervals: specification of the time interval, in seconds, between the triggering of each scheduled task, such as recording or messaging. (4) Model: the detection threshold, representing the minimum confidence score required for an AI model's detection to be transmitted to a remote server. (5) Saving filters: criteria for the permanent storage of audio recordings, including time-based filters (start and end times during which saving is allowed) and a detection threshold, ensuring that only recordings containing detections exceeding the specified confidence are saved. Table D.1 provides a detailed breakdown of the specific settings used in the deployments described in the main text.

Table D.1: Configuration of deployed acoupi programmes. Key settings used to deploy two `acoupi` systems *acoupi_birdnet* for bird monitoring and *acoupi_batdetect2* for bat monitoring during November 2024 at the People and Nature Lab Garden in London, UK. These settings control audio capture, recording schedules, task intervals, and model detection thresholds.

	Programme	
Parameter	*acoupi_birdnet*	*acoupi_batdetect2*
Microphone		
Device	UAC 1.0 Microphone & HID-Mediak	UltraMic 250K 16 bit r4
Samplerate (Hz)	44100	250000
Audio channels	1	1
Recording		
Duration (s)	9	3
Schedule start	00:00:00	17:00:00
Schedule end	23:59:59	07:00:00
Task intervals (s)		
Recording	10	10
Messaging	60	60
Heartbeat	3600	3600
Summarise	3600	3600
Model		
Detection threshold	0.4	0.4
Saving filters		
Start time	06:00:00	17:00:00
End time	22:00:00	07:00:00
Detection Threshold	0.4	0.4

D.2 Detection results

Table D.2: Bat detections by BatDetect2 with acoupi. Summary of the detections by BatDetect2 during a one-month deployment of `acoupi` in November 2024 at the People and Nature Lab Garden, London, UK. For each bat species, the maximum detection score, the total number of detections, and the number of detections with scores above 0.4 and 0.85 are displayed. Rows are sorted by the maximum detection score in descending order. In the UK, bats are typically hibernating during November.

		Detection Counts		
Species	**Max Score**	Total	Score > 0.4	Score > 0.85
Nyctalus leisleri	0.613	1,604,336	170	0
Plecotus austriacus	0.490	205,073	1	0
Pipistrellus nathusii	0.465	113,516	2	0
Pipistrellus pipistrellus	0.403	658,435	1	0
Myotis nattereri	0.387	9,722	0	0
Pipistrellus pygmaeus	0.362	208,803	0	0
Nyctalus noctula	0.348	240,926	0	0
Plecotus auritus	0.303	50,726	0	0
Rhinolophus ferrumequinum	0.286	6,438,263	0	0
Eptesicus serotinus	0.270	1,787,009	0	0
Myotis alcathoe	0.247	38	0	0
Rhinolophus hipposideros	0.197	7,952,268	0	0
Barbastellus barbastellus	0.181	36,023	0	0
Myotis bechsteinii	0.138	2,019	0	0
Myotis daubentonii	0.092	97,757	0	0
Myotis mystacinus	0.070	8,344	0	0
Myotis brandtii	0.022	3,650	0	0

Table D.3: Top detections by BirdNET with `acoupi`. Summary of the top 20 classes detected by BirdNET during a one-month deployment of `acoupi` in November 2024 at the People and Nature Lab Garden, London, UK. For each class, the maximum confidence score, the total number of detections, and the number of detections with confidence scores above 0.4 and 0.85 are displayed. BirdNET was run without geographical filtering, asterisks (*) indicate species known to not occur in the UK. Rows are sorted by the maximum detection score in descending order.

		Detection Counts		
Class	**Max Score**	Total	Score > 0.4	Score > 0.85
Siren	0.999	1382	911	273
Redwing	0.997	313	171	50
Peregrine falcon	0.997	74	40	12
Fireworks	0.997	782	491	175
Eurasian magpie	0.997	707	613	308
White wagtail	0.995	96	74	31
European robin	0.994	813	434	38
Broad-winged hawk*	0.994	164	102	25
Eurasian woodcock*	0.993	8	3	1
Cape May warbler*	0.991	4	2	1
Engine	0.989	7955	3151	73
Egyptian goose	0.987	2	2	2
Little ringed plover	0.982	11	5	1
Malabar whistling thrush*	0.982	20	9	1
Eurasian bittern	0.980	25	17	4
Eurasian wren	0.977	1308	844	61
Arizona toad*	0.976	78	47	10
Belted kingfisher*	0.975	34	23	4
Barn owl	0.973	36	23	1
Yosemite toad*	0.969	27	14	2
Tawny owl	0.969	64	27	4