BirdNET-Annotator: AI-Assisted Strong Labelling of Bird Sound Datasets

Bengt Lüers[1,2], **Patricia P. Serafini**[3,4], **Ivan B. Campos**[3,5], **Thiago S. Gouvêa**[1,2], **Daniel Sonntag**[1,2]
[1]Interactive Machine Leraning, German Research Center for Artificial Intelligence (DFKI), Oldenburg, Germany
[2]Applied Artificial Intelligence, Carl von Ossietzky University of Oldenburg, Germany
[3]National Center for Wild Bird Conservation and Research (CEMAVE), Chico Mendes Institute for Biodiversity Conservation (ICMBio), Brazil
[4]Universidade Federal de Santa Catarina (UFSC), Brazil
[5]Departamento de Biologia Geral, Universidade Federal de Minas Gerais (UFMG), Brazil
{bengt.lueers, thiago.gouvea, daniel.sonntag}@dfki.de
{patricia.serafini, ivan.campos}@icmbio.gov.br

Abstract

Monitoring biodiversity in biosphere reserves is challenging due to the vast regions to be monitored. Thus, conservationists have resorted to employing passive acoustic monitoring (PAM), which automates the audio recording process. PAM can create large, unlabeled datasets, but deriving knowledge from such recordings is usually still done manually.

Machine learning enables the detection of vocalizations of species automatically, allowing summarizing the biodiversity in an area in terms of species richness. While pre-trained neural network models for bird vocalization detection exist, they are often not-reliable enough to do way with the need for manual labeling of audio files.

In this paper, we present BirdNET-Annotator, a tool for AI-assisted labeling of audio datasets co-developed by ecoacoustics and ML experts. BirdNET-Annotator runs in the cloud free of charge, enabling end users to scale beyond the limitations of their local hardware. We evaluated the performance of our solution in the context of its intended workflow and found a reduction in annotation times. While our results show that our application now meets the user requirements, there are still opportunities to seize for additional performance and usability improvement.

Our application illustrates how large, pre-trained neural models can be integrated into the workflow of domain experts when packaged in a user-friendly manner. We observe that although our solution adds a step to the preexisting workflow, the overall annotation speed is significantly improved. This hints at further improvement to be realized in the future by consolidating more steps of the workflow into fewer tools.

1 Introduction

Passive acoustic monitoring (PAM) has become widely used in ecological research, as it is the most scalable data acquisition scheme (Sugai et al. 2018; Tuia et al. 2022). This method involves surveying and monitoring wildlife and environments using sound recorders (acoustic sensors) deployed in the field for extended periods to record acoustic data on a specified schedule. In contrast to active recording, PAM uses omnidirectional microphones, which capture sounds in any direction with equal sensitivity. Use cases for passive acoustic monitoring data include monitoring wildlife reserves, detecting poaching and observing not previously described species.

Machine learning (ML) has proven to be a powerful tool that can achieve great performance in virtually every data analysis task, thus enabling scalable automation for many knowledge work tasks. In contrast to classical programming, where the system behavior is derived from a bottom-up data model, machine learning derives the underlying structure from examples of the data during training, before artificial neural models can be used to make predictions. While machine learning models can be trained on unlabeled data to learn the data structure, associating meaning to data points requires labelled datasets, in which each example is labelled with its meaning in the language of the domain in question (Serrano 2021).

While machine learning systems for automated audio data analysis exist, their applicability is limited to datasets with strong labels (Cobos et al. 2022). While passive acoustic monitoring enables collecting large amounts of data as required by machine learning, labelling the data can quickly become the bottleneck in building datasets. Common approaches to the challenge labelling large datasets poses include crowdsourced, where anonymous users of a service contribute labels to examples. Such libraries include the Macaulay library[1], Xeno-Canto[2], and iNaturalist[3]. While the approach of spreading the labelling workload to many people works, their time is still valuable and assistance hence welcome. To address this need, we propose an automated tool which can assist in labelling audio files for the creation of labelled datasets from passive acoustic monitoring data.

2 Related Work

Birds play a crucial role in soundscape ecology due to their prevalence in the recorded biophony. As birds feed on insects, nectar, and fruits, they are also part of many food chains, making their species richness and abundance good proxies for observing ecosystems (Xie et al. 2023). Solutions for detecting birds from their sounds have advanced from classical pattern recognition to deep learning, significantly improving automatic recognition, e.g. (Potamitis 2016; Eichinski et al. 2022). Current deep learning mod-

[1] https://www.macaulaylibrary.org/

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

[3] https://inaturalist.org

els for bird vocalization detection can discern more than 6,000 species, a significant portion of both the 10,000 described and the estimated 20,000 total bird species (Kahl et al. 2021), which demonstrates both the power of this approach and the need for more data to increase the species coverage.

Audio data labeling tools play a crucial role in streamlining the labeling process and ensuring the accuracy of labeled data, thereby impacting the overall efficiency and accuracy of a machine learning pipeline. A common workflow for labelling audio files is to open the audio file in a desktop application to play it back and annotate regions of its spectrogram representation with the relevant labels. A common tool for this workflow is the open-source application Audacity, which provides a label track for annotations in the time-domain [4]. Extensions to the functionality existing in Audacity like caching the boundaries of selections, automatically inferring binary labels from one another have been proposed as a means of increasing usability. (Li, Burgoyne, and Fujinaga 2006). Also, replacing the audio editor Audacity with a wholly new tool that is specialized for the labelling has been suggested as a more thorough solution (Gibbons et al. 2023). Label Studio is a data labeling platform that has a similar feature set for audio, but also built-in automatic classification tools [5].

Both in the interest of usability and in the interest of model performance it is advisable to identify a narrow context of use and domain of audio to consider. In this work, we propose therefore to add a web tool to the labelling workflow, which employs state-of-the-art machine learning models for automating the labelling step of building passive-acoustic bird sound datasets.

3 Methods

We connected as computer science experts with domain experts to form a consortium which could both identify and build the tool needed for leveraging machine learning for improving the labelling workflow. To guide the process of developing the tool iteratively, we followed Co-design, which refers to a process by which practitioners and researchers come together to either adapt or create and test new materials, exhibits, programs, or technology tools (Mitchell et al. 2015). In our case, the taxonomy experts can be considered actual end users of our proposed application, as they are labelling files from datasets with gold-standard quality on a regular basis. The computer science researchers had been working with passive acoustic monitoring before and most recently built an application for automatic classification of bird song sound snippets, so they were also already familiar with the relevant technologies.

To bring knowledge from both taxonomy and computer science experts together, we conducted a series of semi-structured interviews, which were aimed at understanding the current workflow, its context of use, and the persistent pain points. It turned out that the taxonomists are labelling audio files in sessions of about one or two hours once to twice a week, as part of their jobs, in their offices at university or home. Next, we built a first version of a tool addressing the identified issues within the constraints of the identified context. Installing the often complex machine learning applications may constitute a hurdle to some end users. We therefore designed the tool as a web service that can be accessed through any web browser. The domain experts used the tool for one of their weekly annotation sessions and took notes on their observations, which were discussed with the computer science experts in a following meeting. The user's feedback was considered in a revision of the software artifact. This lead to the addition of an integration with Xeno-Canto, which can be used as a ground-truth source for reference sounds of each detected species. Finally, we conducted a simple user study where the domain expert would label some files both with and without our tool, yielding some promising quantitative results.

4 Implementation

Packaging a complex machine learning application in a way that can easily be deployed to a client computer is still a challenge from a software engineering perspective. This is rooted in the machine learning applications usually being written in Python, which does not provide standard tools for packaging applications. We explored third-party solutions like PyInstaller[6], which has been used with success for machine learning applications before, e.g. in the BirdNET-Analyzer[7]. However, this solution did not yield a deployable package for us. Instead, we choose to package our application as an OCI container image[8] using Google Kaniko[9] and publish it on Docker Hub[10]. This allowed us to deliver the application as a Docker image to a Hugging Face Space for easy and free perpetual hosting.[11] Our application is designed to be used interactively, so responses from the machine learning model usually take about 1 minute on 1-minute input files. Since our Hugging Face Space can easily be forked using the methods of the web interface of Hugging Face, anyone can create a copy for themselves and attach compute resources from Hugging Face's catalog[12] to it, should that be necessary.

Our Docker image is build by GitLab continuous integration[13]. It is based on Ubuntu 22.04[14], installs Python 3.11[15]. Python runtime packages include the BirdNET-Analyzer wrapper birdnetlib[16], the full-stack web toolkit Gradio[17], the

[4] https://manual.audacityteam.org/man/label_tracks.html

[5] https://labelstud.io/

[6] https://pyinstaller.org/en/stable/

[7] https://github.com/kahst/BirdNET-Analyzer

[8] https://github.com/opencontainers/image-spec

[9] https://github.com/GoogleContainerTools/kaniko

[10] https://hub.docker.com/repository/docker/bengt/birdnet-annotator

[11] https://huggingface.co/spaces/Bengt0/BirdNET-Annotator

[12] https://huggingface.co/docs/hub/spaces-gpus

[13] https://docs.gitlab.com/ee/ci/

[14] https://ubuntu.com/

[15] https://www.python.org/

[16] https://pypi.org/project/birdnetlib/

[17] https://pypi.org/project/gradio/

audio processing library librosa[18], the interactive plotting library Plotly[19], the audio resampling library resampy[20], and the deep learning framework TensorFlow[21].

5 Results

Figure 1 shows the finished application from a user's perspective. We made sure to handle all possible implications of interacting with the application so that it can be used in any order of the possible interaction steps. The intended interaction flow starts at the top where the user specifies the dataset to work on either by selecting from a dropdown of predefined locations or by entering the coordinates of the recording locations. This is necessary, as the BirdNET system filters its outputs using the eBird database[22] of local bird species. Next, the users select a recording of the dataset to work from their local machine, which in turn gets uploaded to the server. Note that this interaction scheme allows users to quickly iterate through the audio files of one dataset, which matches the workflow we identified in co-design with the taxonomy experts. The users can configure settings for the inference using the BirdNET Analyzer before a click on the "Detect" triggers the automated analysis. After about 1 minute, the system responds with a list of automatic detections, which can be selected for further inspection. Upon selecting one detection on the left side table, the right side updates to display a spectrogram, an audio player, an interactive label pull-down selection element and a link to Xeno-Canto. Using these, the user can visualize the sound and play it back to detect the bird call themselves. For reference, the user can open the relevant Xeno-Canto page right in the browser they are using the BirdNET-Annotator. The user can select another label from the searchable drop-down or enter a free-text label of their own. Once the user is happy with a label, they can click "Confirm" to add it to the output file. This process can be repeated until the high-confidence detections on this file are exhausted. Finally, the label track file can be downloaded to the user's client computer by clicking "Download". The downloaded file can be readily imported into Audacity for fine-tuning and completing the labels.

We performed a preliminary evaluation of the resulting application using a quantitative user study. An expert labelled 46 minute-long audio files, denoting both species identity and precise temporal boundaries of each sound event, using Audacity either with or without our BirdNET-Annotator application as a preliminary stage. Files were randomised so that difficulty rising and falling with the day and night cycle cannot skew results. The annotation effort took place over the course of several days, so that form-of-the-day effects should be eliminated. We find that the expert could label the files significantly faster when using BirdNET-Annotator than with Audacity alone (reduction from 648 ± 301 to 412 ± 109 seconds, mean ± standard deviation; $p = 0.0012$, Student's t-test). This means BirdNET-Annotator saved an average of 3min and 55s per file, or 36.3% of the time it would normally take using the standard workflow. This observation is in line with the positive subjective response from the taxonomy experts, who report higher efficiency when using BirdNET-Annotator.

6 Conclusion and Future Work

In this paper, we introduced BirdNET-Annotator, a web-based tool that employs large, pretrained artificial neural models to semiautomatically annotate bird calls and bird songs in audio data files provided by the user. We discussed how we developed, built, and hosted BirdNET-Annotator as well as how we evaluated it. BirdNET-Annotator has proven to be easily accessible, usable by domain experts, and to improve user performance as measured in labelling time.

In our exchanges between machine learning and domain experts, it became clear that automation of the tedious work of instance labelling is vital. We could show that the increased automation pays off in terms of labeling time.

While the introduction of an AI-powered element proved to be advantageous, it also introduced the need for the user to switch between two applications, namely BirdNET-Annotator for curated model-based annotation, and Audacity for refining temporal boundaries of prediction as well as correction of false negatives. The need to switch between applications may represent a hurdle, and it would be desirable to combine all functionalities in one platform. We explored building the labelling functionality as in Audacity right into the spectrogram plot of the BirdNET-Annotator, but hit a wall due to our choice for Gradio as a library for generating the front end. Specifically, in its current version, Gradio does not expose functionality necessary for interacting with plots, writing annotations and listening for user interactions with them. While one could in principle implement the functionality oneself, we opted for not pursuing that path due to the complexity of the Plotly element we used for this plot. As a viable alternative given the tools available today, the application could be reimplemented in a framework which supports this central requirement. Further, we explored the possibility of rewriting BirdNET-Annotator in the plotly-aware full-stack web framework Dash[23] and demonstrated a successful combination of the spectrogram plot with the interactive labelling component. Given the great potential of AI-assisted tools for the taxonomy domain, this seems to be worth exploring more in future iterations of the annotation tool.

In conclusion, our study demonstrates the successful integration of large, pre-trained neural models within the workflow of specialized domain experts, showcasing the potential of interactive, user-friendly applications. It turns out that bird sound datasets form an excellent application field for explorative AI-assisted labelling, a precondition for the effective application of supervised machine learning algorithms. The present work may offer a template for broader applications, highlighting the potential of interactive machine learning across diverse domains.

[18] https://pypi.org/project/librosa/

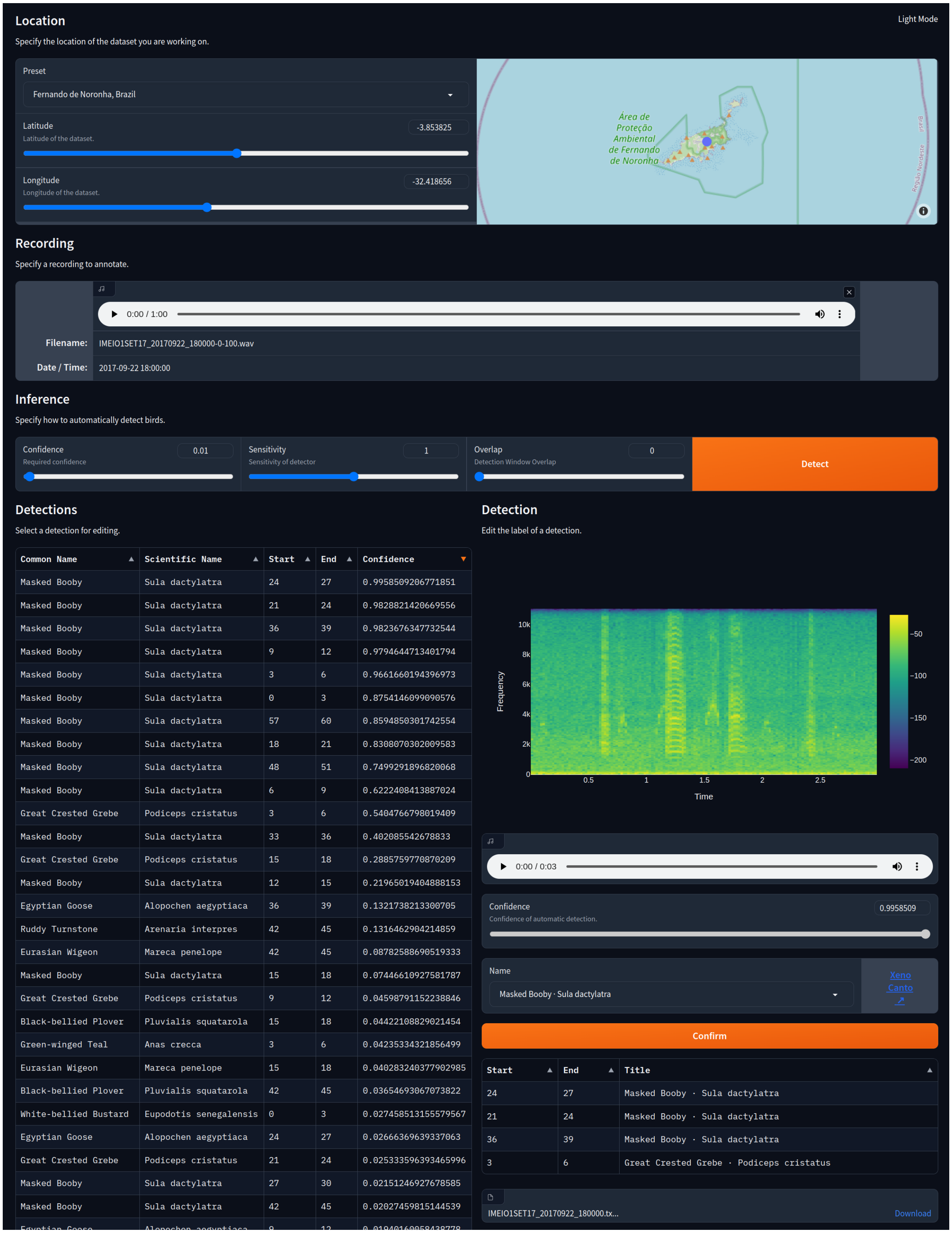
[19] https://pypi.org/project/plotly/

[20] https://pypi.org/project/resampy/

[21] https://pypi.org/project/tensorflow/

[22] https://ebird.org/

[23] https://plotly.com/dash/



Location

Specify the location of the dataset you are working on.

Preset: Fernando de Noronha, Brazil

Latitude: -3.853825
Latitude of the dataset.

Longitude: -32.418656
Longitude of the dataset.

Recording

Specify a recording to annotate.

Filename: IMEIO1SET17_20170922_180000-0-100.wav

Date / Time: 2017-09-22 18:00:00

Inference

Specify how to automatically detect birds.

Confidence: 0.01
Required confidence

Sensitivity: 1
Sensitivity of detector

Overlap: 0
Detection Window Overlap

Detections

Select a detection for editing.

Common Name	Scientific Name	Start	End	Confidence
Masked Booby	Sula dactylatra	24	27	0.9958509206771851
Masked Booby	Sula dactylatra	21	24	0.9828821420669556
Masked Booby	Sula dactylatra	36	39	0.9823676347732544
Masked Booby	Sula dactylatra	9	12	0.9794644713401794
Masked Booby	Sula dactylatra	3	6	0.9661660194396973
Masked Booby	Sula dactylatra	0	3	0.8754146099090576
Masked Booby	Sula dactylatra	57	60	0.8594850301742554
Masked Booby	Sula dactylatra	18	21	0.8308070302009583
Masked Booby	Sula dactylatra	48	51	0.7499291896820068
Masked Booby	Sula dactylatra	6	9	0.6222408413887024
Great Crested Grebe	Podiceps cristatus	3	6	0.5404766798019409
Masked Booby	Sula dactylatra	33	36	0.402085542678833
Great Crested Grebe	Podiceps cristatus	15	18	0.2885759770870209
Masked Booby	Sula dactylatra	12	15	0.21965019404888153
Egyptian Goose	Alopochen aegyptiaca	36	39	0.1321738213300705
Ruddy Turnstone	Arenaria interpres	42	45	0.1316462904214859
Eurasian Wigeon	Mareca penelope	42	45	0.08782588690519333
Masked Booby	Sula dactylatra	15	18	0.07446610927581787
Great Crested Grebe	Podiceps cristatus	9	12	0.04598791152238846
Black-bellied Plover	Pluvialis squatarola	15	18	0.04422108829021454
Green-winged Teal	Anas crecca	3	6	0.04235334321856499
Eurasian Wigeon	Mareca penelope	15	18	0.040283240377902985
Black-bellied Plover	Pluvialis squatarola	42	45	0.03654693067073822
White-bellied Bustard	Eupodotis senegalensis	0	3	0.027458513155579567
Egyptian Goose	Alopochen aegyptiaca	24	27	0.02666369639337063
Great Crested Grebe	Podiceps cristatus	21	24	0.025333596393465996
Masked Booby	Sula dactylatra	27	30	0.02151246927678585
Masked Booby	Sula dactylatra	42	45	0.02027459815144539
Egyptian Goose	Alopochen aegyptiaca	9	12	0.01940160058438778

Detection

Edit the label of a detection.

Confidence: 0.9958509
Confidence of automatic detection.

Name: Masked Booby · Sula dactylatra

Start	End	Title
24	27	Masked Booby · Sula dactylatra
21	24	Masked Booby · Sula dactylatra
36	39	Masked Booby · Sula dactylatra
3	6	Great Crested Grebe · Podiceps cristatus

IMEIO1SET17_20170922_180000.tx...

Figure 1: A screenshot of the BirdNET-Annotator application interface. The user inputs a sound file, indicating the geographical location where it was recorded. Sliders are used to set parameters such as confidence- and sensitivity-thresholds, as well as desired overlap between contiguous time windows. Inference with the pre-trained BirdNET model is triggered through the large, bright-orange "Detect" button. Individual detections are listed on the bottom-left UI element (common and scientific names, segment start and end times, and confidence level associated with the prediction). The domain expert can evaluate the prediction by listening to the sound segment, as well as visualising it as a spectrogram. A link to the predicted species' page on the reference platform Xeno-Canto is provided for quick comparison. If needed, the user can correct the prediction by assigning a different species label to the sample. Finally, the user can confirm the label and move on to the next sample.

Acknowledgements

This research is part of the Computational Sustainability & Technology project field[24], and has been supported by the Ministry for Science and Culture of Lower Saxony (MWK), the Endowed Chair of Applied Artificial Intelligence, Oldenburg University, and DFKI.

References

Cobos, M.; Ahrens, J.; Kowalczyk, K.; and Politis, A. 2022. An overview of machine learning and other data-based methods for spatial audio capture, processing, and reproduction. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1): 1–21.

Eichinski, P.; Alexander, C.; Roe, P.; Parsons, S.; and Fuller, S. 2022. A convolutional neural network bird species recognizer built from little data by iteratively training, detecting, and labeling. *Frontiers in Ecology and Evolution*, 10: 133.

Gibbons, A.; Donohue, I.; Gorman, C.; King, E.; and Parnell, A. 2023. NEAL: an open-source tool for audio annotation. *PeerJ*, 11: e15913.

Kahl, S.; Wood, C. M.; Eibl, M.; and Klinck, H. 2021. BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, 61: 101236.

Li, B.; Burgoyne, J. A.; and Fujinaga, I. 2006. Extending Audacity for Audio Annotation. In *ISMIR*, 379–380.

Mitchell, V.; Ross, T.; May, A.; Sims, R.; and Parker, C. J. 2015. Empirical investigation of the impact of using codesign methods when generating proposals for sustainable travel solutions.

Potamitis, I. 2016. Deep learning for detection of bird vocalisations. *arXiv preprint arXiv:1609.08408*.

Serrano, L. 2021. *Grokking Machine Learning*. Simon and Schuster.

Sugai, L. S. M.; Silva, T. S. F.; Ribeiro, J., José Wagner; and Llusia, D. 2018. Terrestrial Passive Acoustic Monitoring: Review and Perspectives. *BioScience*, 69(1): 15–25.

Tuia, D.; Kellenberger, B.; Beery, S.; Costelloe, B. R.; Zuffi, S.; Risse, B.; Mathis, A.; Mathis, M. W.; van Langevelde, F.; Burghardt, T.; et al. 2022. Perspectives in machine learning for wildlife conservation. *Nature communications*, 13(1): 792.

Xie, J.; Zhong, Y.; Zhang, J.; Liu, S.; Ding, C.; and Triantafyllopoulos, A. 2023. A review of automatic recognition technology for bird vocalizations in the deep learning era. *Ecological Informatics*, 73: 101927.

[24] https://cst.dfki.de/