Bioacoustic Monitoring of Endangered Bird Population

B.Sai Kumar[1], D. Madhan[2], K. Harischandraprasad[3], M. Yellamma[4]
[1,2,3,4]*Member, Dept of Computer Science and Engineering, Sreenidhi Institute of Science and Technology*

Abstract: **Extinction is a natural part of evolution and occurs for many reasons. Understanding the causes of extinction and how birds help overcome these threats is before more species disappear, as our world changes and not all birds can change with it. It's a great way to promote bird protection. Today, birds are becoming extinct, with significant increases in numbers over the last decade, affecting the entire food chain. Population monitoring is used to understand how native birds respond to environmental changes and conservation activities. However, many of the island's birds live in isolation in hard-to-reach highland habitats. Physically monitoring the presence of birds is a complex process and requires an automated process. Recent advances in machine learning have made it possible to use extensive training data to automatically identify bird calls of common species. However, developing such models for rare and endangered species remains a challenge.**

***Keywords*— Bioacoustic Monitoring, convolutional neural network, audio classification, EDA.**

I. INTRODUCTION

Birds play a crucial role in a variety of habitats. They serve an important role in pest control, pollination, and preserving island ecology. Birds are also beneficial to humans in a variety of ways, including providing food and fertilizer in agricultural contexts.

Bird conservation demands a thorough understanding of their spatiotemporal occurrence and dispersal patterns. Passive acoustic monitoring (PAM) has become an important technology for collecting data on birds at ecologically relevant sizes over the last decade. However, these PAM attempts generate large datasets, which make full analysis difficult. To progress the subject of bird conservation, better and fully automated acoustic analytic frameworks are required.Recent advances in the development of machine listening algorithms for identifying animal vocal sounds have increased our ability to evaluate long-term acoustic recordings in a comprehensive manner. However, producing analysis outputs with high precision and recall remains difficult, especially when targeting a large number of species at once. One of the most difficult issues in the field of auditory event detection and identification is bridging the domain gap between high-quality training samples and noisy test samples.

Algorithmic sound identification has seen a steady rise in popularity in recent years. The popularity of deep learning and the various types of neural networks has opened up a new way of treating classification problems that has yet to be explored.

One of the most extensively utilized applications in Audio Deep Learning is sound classification. It entails learning to classify sounds and predict which group they belong to. This type of challenge can be used in a variety of situations, such as classifying music clips to determine the genre of music, or classifying short utterances by a group of speakers to determine the speaker based on the voice.Various classifiers were employed to classify the crossing rate, pitch, and frame information used in speech recognition applications.In recent years, the terms Spectrogram image features (SIF), Stabilized auditory image (SAI), and Linear prediction coefficients (LPC) have become increasingly popular.

In noisy environments, SIF generates sound waves, yielding more accurate outcomes. High-pressure and low-pressure zones move through a medium to create sound waves. Every distinguishable sound has a unique pattern formed by such high- and low-pressure zones. Wavelength, frequency, wave speed, and time periods are all features of these waves. These qualities are used to classify the sounds in the same way that people do.

Deep learning methods were utilized to accomplish feature extraction and classification from spectrogram images since they are a visual representation of a signal's frequency spectrum. Sound signals are less common, have a weak location, and produce a variety of spectrogram

patterns. CNN, on the other hand, is gaining favor in computer vision and audio processing, where it is insensitive to pattern position on the output spectrogram image and is acknowledged as a good technique for categorizing spectrogram image characteristics.A convolutional layer is the initial layer in CNN, and it attempts to learn the image's underlying information. The pooling layer follows, which attempts to minimize the feature map's dimensionality. The feature map comes from the convolutional layer, which is passed to the pooling layer. Depending on the size and complexity of the dataset, there may be numerous sets of convolutional and pooling layers. A classification or prediction layer is the final layer of a convolutional neural network. The success of convolutional neural networks is based on three important characteristics. These are local receptive fields. Divided weighting and spatial subsampling. Local receptive fields mean that neuronal responses are affected by specific 2D parts of the image. Shared weights are a plus point for convolutional neural networks. This is because using these weights reduces the total number of parameters in the network. Subsampling is used to reduce the resolution of feature maps. This solves the problem of distortion and shifts in the final output. In this way the CNN's are used widely for sound classification.

II. METHODS

In this research, we have used the deep convolutional neural networks to identify the bird calls in sound space.

A. Data

The BirdCLEF 2022 challenge featured one of the largest collections of soundscape recordings from rare birds in Hawaii. With regard to real-world use cases, labels and metrics were chosen to reflect the vast diversity of bird vocalizations and variable ambient noise levels in omnidirectional recordings. We will develop a model that can process continuous audio data and then identify the species.

B. Spectrograms

A spectrogram is a graphical representation of the spectrum of frequencies of a signal over time. When spectrograms are used to analyze audio signals, they are sometimes called sonography, voiceprints, or voicegrams. The data may be displayed in a waterfall form when they are plotted in 3D.A graph can often be used to visualize data in two dimensions, with time on one axis and frequency on the other. A third dimension, amplitude, can be represented by color or intensity.

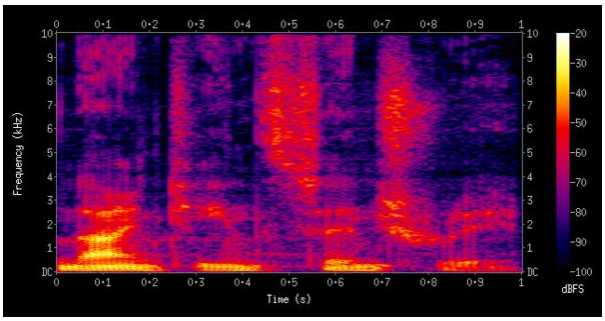

Fig 1: A spectrogram

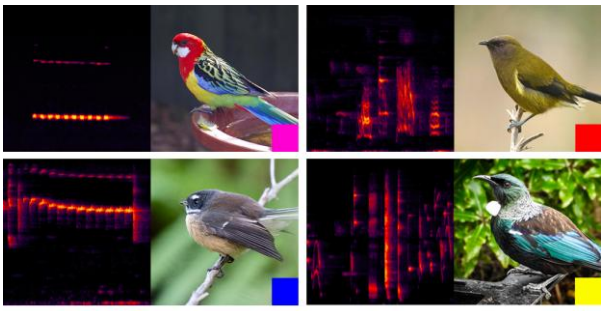

Fig 2: The spectrograms generated for different bird calls

To create a spectrogram, a time-domain signal is divided into shorter segments of equal length. The fast Fourier transform (FFT) is then applied to each segment. The spectrum is a graph of the spectral content on each segment. The Frame Count parameter determines the number of Fourier transforms used to create the spectrogram and, as a result, the number of individual time signals that are split into independent frequency components.

C. Convolutional Neural Network

Convolutional networks, also known as convolutional neural networks, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology. Convolutional networks have been tremendously successful in practical applications. The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks

that use convolution in place of general matrix multiplication in at least one of their layers.

For a typical input audio that we have, it will be preprocessed and the spectrogram is generated as a next step.Then we will pass the spectrogram to the CNN so that it will identify and classify the input audio.

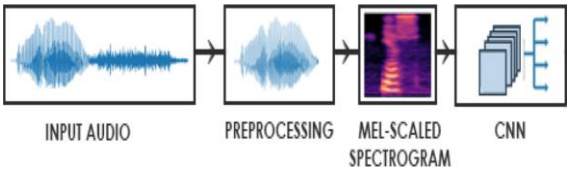

Fig 3: Steps involved in audio classification

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

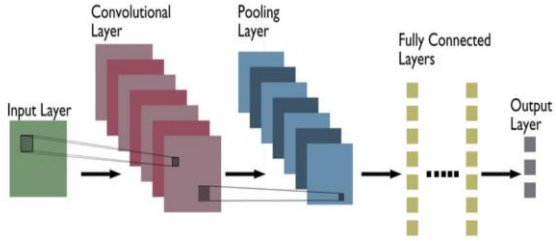

Fig 4: CNN Architecture

1. Convolutional layer

The convolutional layer performs the convolution operation. In most general form, convolution is an operation on two functions of a real valued argument. A convolution is the use of a filter to process an input signal. This will activate the input signal, depending on the filter type and settings. Applying the same filter to an input repeatedly produces a map of activations called a feature map, which reveals the locations and strength of a detected feature in the input, such as an image.

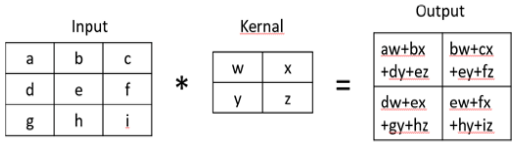

Fig 5: Convolution Operation

As the convolution operation happens repeatedly, there is a problem of the image being shrunk and so it becomes small. We add an extra layer throughout the circumference of our original image to retain its features without being shrunk, by convention we pad with zeros.

Reasons for why we go for padding :

1) As the convolution operation happens repeatedly there is a problem of our input image being shrinked , so its original form will be lost till it reaches the last step.

2) When a filter is convolved over an input image , then only once the top left corner occurs and pixel in the middle would come several times, this leads to throwaway information from corners.

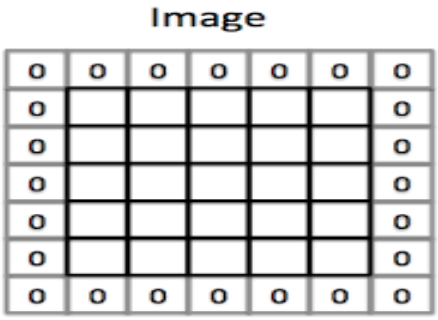

Fig 6: Image padded with zeros

Stride is a component of convolutional neural networks, which are neural networks that are specifically designed to reduce the size of images and videos. Stride is a parameter of the neural network's filter that controls how much movement is displayed in an image or video.

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable size.

Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means that every output unit interacts with every input unit. Convolutional networks, however, typically have sparse interactions. This is accomplished by making the kernel smaller than the input.

Parameter sharing refers to using the same parameter for more than one function in a model. In a traditional neural net, each element of the weight matrix is used exactly once when computing the output of a layer. It is multiplied by one element of the input and then never visited. As a synonym for parameter sharing, one can say that a network has tied weights, because the value of the weight applied to one input is tied to the value of a weight applied elsewhere. In a cnn, each member of the kernel is used at every position

of the input (except perhaps some of the boundary pixels, depending on the design decisions regarding the boundary). The parameter sharing used by the convolution operation means that rather than learning a separate set of parameters for every location, we can learn only one set. In the case of convolution, the particular form of parameter sharing causes the layer to have a property called equivariance to translation. To say a function is equivariant means that if the input changes, the output changes in the same way. With images, convolution creates a 2-D map of where certain features appear in the input. If we move the object in the input, its representation will move the same amount in the output. This is useful when we know that some function of a small number of neighboring pixels is useful when applied to multiple input locations. Convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image. Other mechanisms are necessary for handling these kinds of transformations.

2. Pooling layer

In the pooling layer, we use a pooling function to modify the output of the layer further. A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling operation reports the maximum output within a rectangular neighborhood.Other popular pooling functions include the average of a rectangular neighborhood, or a weighted average based on the distance from the central pixel.

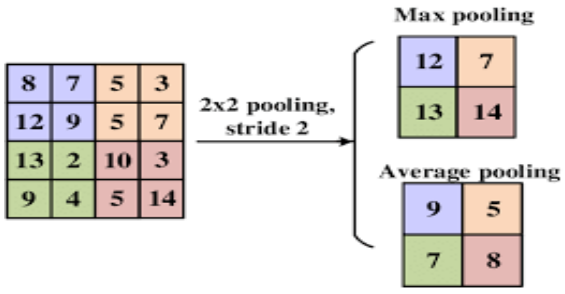

Fig 7: Max and Average pooling

In all cases, pooling helps to make the representation approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change. The use of pooling can be viewed as adding an infinitely strong prior that the function the layer learns must be invariant to small translations. When this assumption is correct, it can greatly improve the statistical efficiency of the network. Pooling over spatial regions produces invariance to translation, but if we pool over the outputs of separately parametrized convolutions, the features can learn which transformations to become invariant to. For many tasks, pooling is essential for handling inputs of varying size. For example, if we want to classify images of variable size, the input to the classification layer must have a fixed size. This is usually accomplished by varying the size of an offset between pooling regions so that the classification layer always receives the same number of summary statistics regardless of the input size.

3. Fully connected layer

The fully connected layer is a dense layer that feeds the outputs of the convolutional layers via one or more neural layers to generate a prediction. In the case of a completely linked layer, all of the elements from the previous layer's features are included in the computation of each element of each output feature, which includes an activation function depending on our business requirements. There is a 'Flatten' layer that sits between the convolutional and fully linked layers. A two-dimensional matrix of features can be flattened into a vector that can be input into a fully connected neural network classifier.

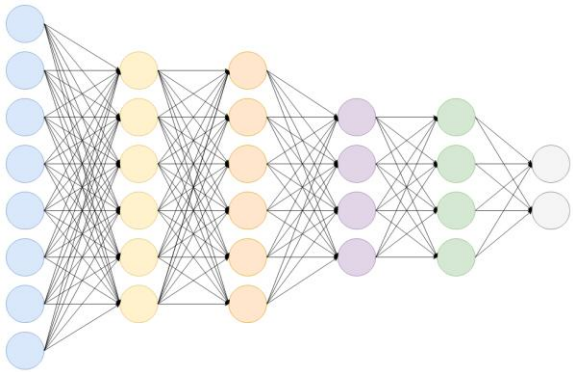

Fig 8: Fully connected network

D. ResNet

In the research community, there is a growing consensus that our network design has to be more complex. However, just adding layers together will not increase network depth. Deep networks are difficult to train due to the well-known vanishing

gradient problem: when a gradient is back-propagated to earlier layers, repeated multiplication can lead the gradient to become infinitely small. As a result, as the network grows larger, its performance becomes saturated, if not drastically degraded. There have been various approaches to dealing with the vanishing gradient problem before ResNet, but none seemed to truly solve the problem. ResNet's main concept is to introduce a "identity shortcut link" that bypasses one or more layers, as indicated in the diagram below:

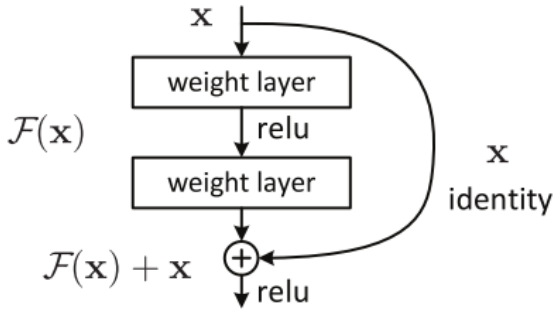

Fig 9: A residual block

E. Loss Function

BCE with logit loss is the loss function used. In this loss, a Sigmoid layer and the BCELoss are combined into a single class. This approach is more numerically stable than using a plain Sigmoid followed by a BCELoss because we use the log-sum-exp method for numerical stability by integrating the operations into one layer.

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

F. Optimizer

The optimizer used is Adam optimizer. Adam is an adaptive learning rate optimization algorithm. The name “Adam” derives from the phrase “adaptive moments”. It is perhaps best seen as a variant on the combination of RMSProp and momentum with a few important distinctions. First, in Adam, momentum is incorporated directly as an estimate of the first-order moment (with exceptional weighting) of the gradient. The most straightforward way to add momentum to RMSProp is to apply momentum to the rescaled gradients. The use of momentum in combination with rescaling does not have a clear theoretical motivation.

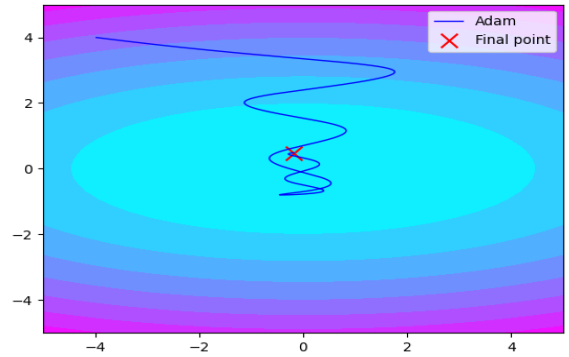

Fig 10: Adam Optimizer movement

Second, Adam includes biased corrections to the estimates of both the first-order moments (the momentum term) and the (uncentered) second-order moments to account for their initialization at the origin. RMSProp also incorporates an estimate of the (uncentered) second-order moment; however, it lacks the correction factor. Thus, unlike in Adam, the RMSProp second-order moment estimate may have high bias early in training. Adam is generally regarded as being fairly robust to the choice of hyperparameters, though the learning rate sometimes needs to be changed from the suggested default.

III. TRAINING

The dataset BirdClef 2022 is trained using the convolutional neural networks and ResNet. The loss function used is BCE with logit loss and the optimizer used is Adam optimizer. The below is the plot generated after training for loss (blue) and validation loss (orange).

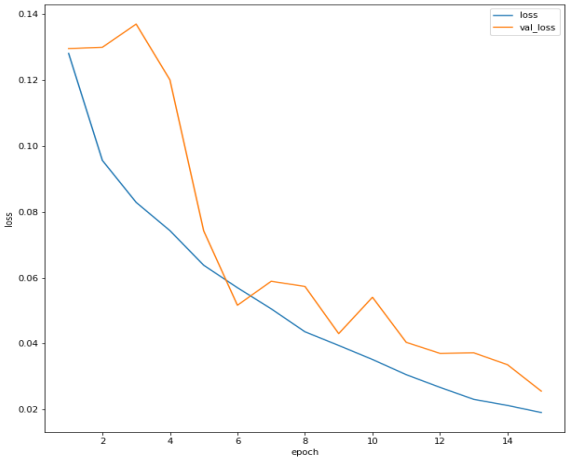

Fig 11: Epoch VS Loss

IV. DEPLOYMENT

The model developed is deployed using Flask. Flask is a compact and lightweight Python web framework that provides essential tools and capabilities for building online applications in Python. Because you can build a web application rapidly using only a single Python file, it allows developers more flexibility and is a more accessible framework for beginning developers. Flask is also extendable, and it doesn't require complicated boilerplate code or a certain directory structure to get started. The interface developed is in the figure below.

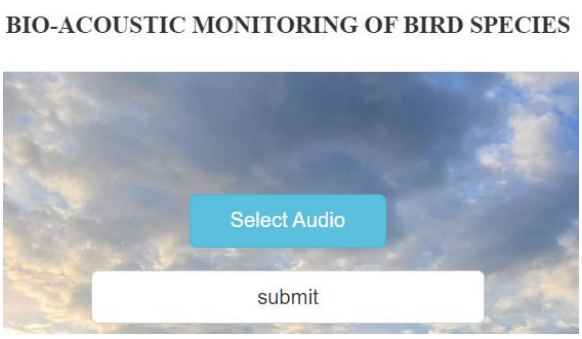

Fig 12: UI developed for uploading audio

V. RESULTS

Upon selecting the audio the model would return its predictions. For a particular audio given the results are as shown in the figure below.

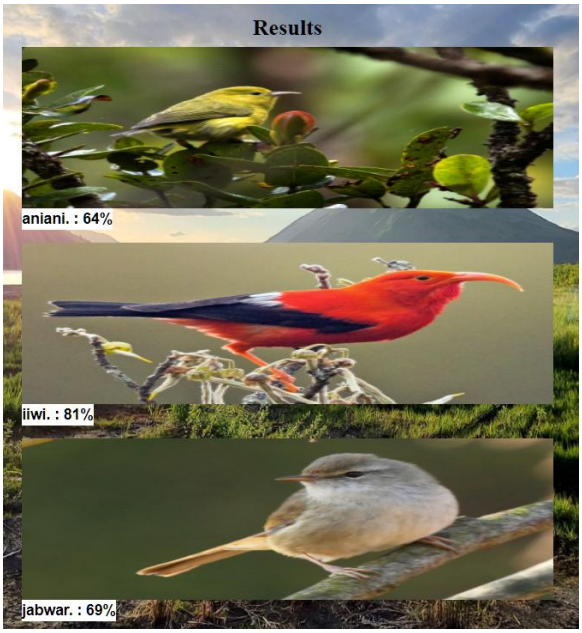

Fig 13: Results for a given audio

VI. CONCLUSION

Hawai'i, known as the "extinction capital of the world," has lost 68 percent of its bird species, with serious ramifications for entire food systems. Population monitoring is used by researchers to learn how native birds respond to changes in the environment and conservation activities. However, many of the islands' remaining birds are isolated in difficult-to-reach high-elevation habitats. Due to the difficulty of physical monitoring, scientists have turned to sound recordings. This method, known as bioacoustic monitoring, could be used to research endangered bird populations in a passive, low-labor, and cost-effective manner. Manual annotation of each recording is the current method for processing huge bioacoustic datasets. This necessitates specialized knowledge and an inordinate amount of time. With enough training data, recent breakthroughs in machine learning have made it possible to automatically identify popular bird songs. However, developing such tools for rare and endangered species like those found in Hawai'i remains difficult.
The K. Lisa Yang Center for Conservation Bioacoustics (KLY-CCB) at Cornell Lab of Ornithology develops and deploys innovative conservation technologies across different ecological scales to inspire and inform animal and habitat conservation. For this competition, KLY-CCB has teamed up with Google Bioacoustics Group, LifeCLEF, the Listening Observatory for Hawaiian Ecosystems (LOHE) Bioacoustics Lab at the University of Hawai'i at Hilo, and Xeno-Canto to collect and interpret sounds in nature. Bird species are identified using machine learning and deep learning techniques. We created a model that can process continuous audio data and then recognise the species acoustically. This will contribute to the advancement of bioacoustics research as well as ongoing efforts to safeguard endangered Hawaiian birds.

VII. FUTURE ENHANCEMENT

In recent years, there has been a continuous development in the field of bird sound detection using machine learning approaches, with most of the work focusing on training diverse neural networks from the start. The following improvements should be explored in order to attain higher accuracies. By fine-tuning the hyperparameters, a more resilient pre-trained convolutional neural network can be created,

allowing for higher accuracies. Furthermore, we propose noise reduction and filtering during the preprocessing stage to eliminate extreme frequencies not present in bird noises.

REFERENCE

[1] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," IEEE Signal Processing Letters, vol. 24, pp. 279–283, 2017.

[2] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Leveraging deep neural networks with nonnegative representations for improved environmental sound classification," IEEE International Workshop on Machine Learning for Signal Processing MLS, 2017

[3] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 25, no. 3, pp. 235–238, Jun 1977

[4] S. Kahl, M. Clapp, W. Hopping, H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, A. Joly, Overview of BirdCLEF 2020: Bird sound recognition in complex acoustic environments, in: CLEF task overview 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece., 2020.

[5] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, BirdNET: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236.

[6] Y. Shiu, K. Palmer, M. A. Roch, E. Fleishman, X. Liu, E.-M. Nosal, T. Helble, D. Cholewiak, D. Gillespie, H. Klinck, Deep neural networks for automated detection of marine mammal species, Scientific reports 10 (2020) 1–12.

[7] V. Ruíz, J. D. Román, J.-Y. Duriaux, The Sounds of Sustainability, 2021.

[8] C. M. Wood, V. D. Popescu, H. Klinck, J. J. Keane, R. Gutiérrez, S. C. Sawyer, M. Z. Peery, Detecting small changes in populations at landscape scales: A bioacoustic site-occupancy framework, Ecological Indicators 98 (2019) 492–507.

[9] C. M. Wood, S. Kahl, P. Chaon, M. Z. Peery, H. Klinck, Survey coverage, recording duration and community composition affect observed species richness in passive acoustic surveys, Methods in Ecology and Evolution 12 (2021) 885–896.

[10] S. Kahl, F.-R. Stöter, H. Glotin, R. Planqué, W.-P. Vellinga, A. Joly, Overview of birdclef 2019: Large-scale bird recognition in soundscapes, in: CLEF working notes 2019, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2019, Lugano, Switzerland. 2019.

[11] N. Murakami, H. Tanaka, M. Nishimori, Bird Call Identification using CNN and Gradient Boosting Decision Trees with Weak and Noisy Supervision, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, 2021.

[12] C. Henkel, P. Pfeiffer, P. Singer, Recognizing bird species in diverse soundscapes under weak supervision, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, 2021.

[13] D. Stowell, M. D. Plumbley, An open dataset for research on audio field recording archives: freefield1010, arXiv preprint arXiv:1309.5275 (2013).

[14] M. V. Conde, N. D. Movva, P. Agnihotri, S. Bessenyei, K. Shubham, Weakly-Supervised Classification and Detection of Bird Sounds in the Wild. A BirdCLEF 2021 Solution, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, 2021.

[15] J.-F. Puget, STFT Transformers for Bird Song Recognition, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, 2021.