Bird Species Classification from Acoustic Data Using a Convolutional Neural Network

Anuradha Ramachandran
University of Washington
Department of Statistics
anuram08@uw.edu

Monica Ramsey
University of Washington
Department of Statistics
mbr03@uw.edu

Abstract

Acoustic monitoring of bird populations is an important tool for ecological, environmental, and conservation research. In recent years, data acquisition and processing methods have been facilitated by advancements in deep learning frameworks, most notably by convolutional neural networks (CNNs). Prior research has found that CNN-based models have demonstrated high performance in bird species identification and classification; however, given the vast number of bird species worldwide, there is a need for models to classify bird species on a global scale. In this paper we present a CNN-based model to identify Eastern African bird species. Our approach involves transforming input data to visual input in the form of a mel-scale spectrogram. We explore two CNN architectures, resulting in a baseline validation accuracies of 74% and 92% respectively. In addition, we explore various hyperparameter tuning methods including modifying mel spectrogram features to find a model that achieves high accuracy.

1. Introduction

In recent decades, acoustic monitoring has become a widely used method for conducting ecological and environmental research [1]. In particular, passive acoustic monitoring (PAM), which entails capturing continuous audio recordings from an environment using remote recording devices, has become a popular approach for monitoring bird populations [1]. As many birds are highly vocal and are generally more easily detectable by sound than by sight, researchers have used audio data to survey biodiversity, assess wildlife conservation efforts, and evaluate changes in avian populations [2, 3].

Within the last decade, a particular interest has grown in applying deep learning techniques for animal species classification [4, 5]. In a related study, Bergler et al. [6] implemented a convolutional neural network (CNN) to classify killer whale vocalizations based on audio input, which was converted to spectrograms using a Fast Fourier Transform (FFT). Similarly, Sprengel et al. [7] applied a six-layer CNN, which was trained on on spectrograms generated by Short Time Fourier Transform of input audio data, for the classification of 999 bird species. LeBien et al. [8] adopt a ResNet-50 model to classify 24 bird and frog species using spectrograms as model input. In addition to the aforementioned models, several variations of CNN architectures have been employed for species identification and have been found to achieve high performance on image recognition and classification tasks [9, 10]. As such, CNNs have been widely accepted as a useful framework for bird species identification [9].

These methods have been further popularized by initiatives including the LifeCLEF Bird Detection Challenge (BirdCLEF), an annual bird species identification competition which has attracted a growing number of participants since its inception in 2014 [11]. One such application of a CNN framework is BirdNET, one of the largest initiatives in bird sound classification. Thanks to openly-available repositories of bird song recordings such as xeno-canto, as well as contributions from "citizen scientists," BirdNET's identification capacity has expanded from 984 species to over 3000 of the world's most common bird species [12].

2. Research Question

Recent studies in avian biodiversity estimate that there are approximately 11,000 distinct bird species worldwide [13]. While BirdNET and similar CNN-based models have been successfully implemented for the classification of a variety of avian species, there remains a need for models that can classify bird species on a global scale. We note that previous versions of the BirdNet model encompassed only bird species prevalent in North America and Europe [12], while the present version does not contain equal representation of species across geographic regions [14].

The organizers of the 2023 BirdCLEF challenge have

partnered with Kenyan conservation organization NATURAL STATE to identify the sounds of Eastern African bird species [15]. To our knowledge, there is currently no existing sound classification model for these species. As such, we seek to develop a model to classify Eastern African bird species based on audio data.

3. Research Methodology

Due to their high performance on audio-based classification tasks, CNN-based models have become one of the most widely-used frameworks for bird sound classification [9, 10]. Accordingly, we will employ a CNN architecture in the present study to train a model for identifying Eastern African bird species. In this study, we will apply a CNN architecture to train a model to identify Eastern African bird species. We first pre-process our data by converting audio input, which is in the form of audio files containing bird call and song recordings of varying lengths, to visual output in the form of spectrograms. Using spectrograms for feature extraction allows us to undertake a multiclass identification problem. Please see Section 4.2.1 for a detailed explanation of our data pre-processing methods.

4. Project Design

4.1. Dataset Information

The dataset for this model was sourced from the BirdCLEF 2023 Kaggle challenge, which contains 16,900 audio recordings of vocalizations from Eastern African bird species. The audio samples, which were of non-uniform length, were in ogg format with a sampling rate of 32kHz. The dataset also included a corresponding file named train-metadata.csv which contained relevant information for each audio recording, including "primary label," which represents the species label for each recording; "secondary label," which denoted any other species labels if there was more than one bird present in the recording, if any; "rating," which denotes the quality of the audio recording; as well as the latitude and longitude coordinates of where the vocalization was recorded. In total there were 264 species, as indicated by the "primary label" in the metadata file.

Upon initial review of the dataset, it was evident that there was class imbalance among species: while some species contained over 500 audio samples, there were numerous species for which there were three or fewer audio samples present in the dataset. Moreover, the quality of the audio recordings ranged from 0 to 5, with increments of 0.5 and was non-uniform across the entire dataset. As class imbalance is known to negatively impact model performance, we implemented a number of techniques to address the unequal representation of species in our dataset. We describe these methods in greater detail in 5.3 and 5.4.

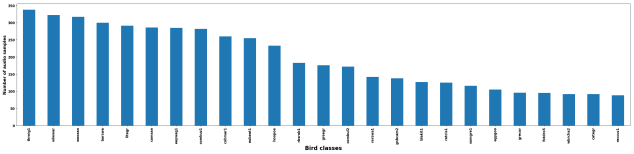

Figure 1. Audio sample distribution for top 25 species

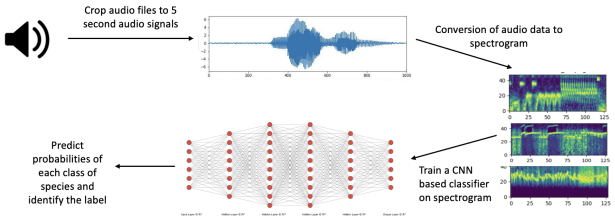

Figure 2. Training process flow diagram

4.2. Model Training Procedure

All experiments in the present study are implemented using Python 3.10.11 and related libraries. For audio preprocessing and spectrogram generation we used librosa version 0.10.0; for model training and validation we used TensorFlow and Keras version 2.12.0, as well as scikit-learn version 1.2.2. For faster computation, we trained our model on Google Colaboratory ("Colab"), which is a cloud-based platform with access to GPU assistance (subject to availability) [16]

For the present study, our performance metrics are accuracy, recall, and precision. We include recall and precision as these metrics are better indicators of model performance for imbalanced datasets. An overview of the steps involved in this project is summarized in Figure[2]. The steps in this project is elaborated in the following subsections.

4.2.1 Data Preprocessing

Due to class imbalance among species and non-uniformity in the audio quality in the training dataset, only audio files from species that had > 88 audio samples with a rating greater than 4 were used for baseline training. A total 4916 audio samples belonging to 25 bird species that met this criteria were used for training and validation of the classifier. Even within this subset of data, class imbalance is evident as shown in Figure 1. As the audio files in the training data are not of uniform size, all the audio files were cropped to 5 second clips. Since CNN architecture is shown to work well for image classification data with uniform input size, the audio clips are transformed into a gray-scale mel spectrograms of uniform size using the librosa library (Version 0.10.0) [17]. Mel spectrograms were generated using a sampling rate of 32,000 Hz with low pass and high pass filters at 500 Hz and 12.5 kHz respectively. This process gener-

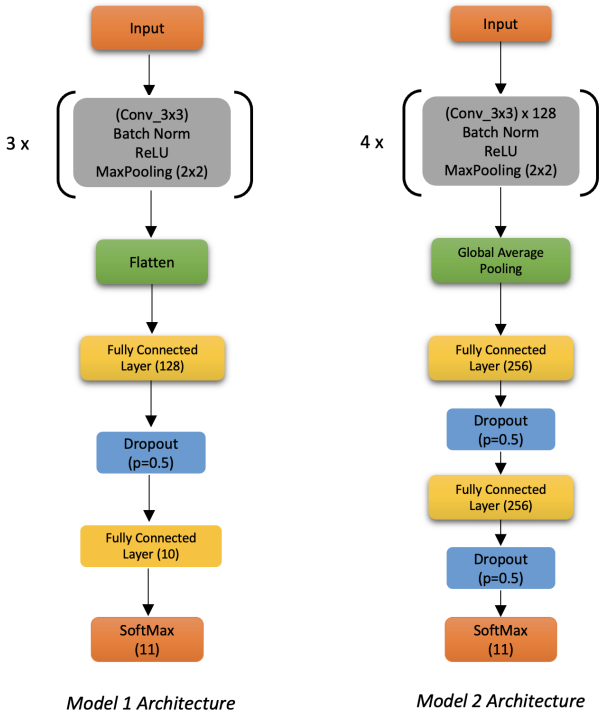

Figure 3. CNN architectures for BirdCLEF

ated a total of 45,575 mel spectrograms corresponding to 25 species. Each of the mel spectrograms were then flattened to obtain the dimensions (Height x Width x 1).

```
def get_mel spectrogram(audio_file, label, outputdir):
    load the audio_file
    trim the dead noises from the audio file

    for audio_chunk in range(0, length(audio_file), sample_length):
        crop the audiodata into 5 second chunks
        convert this audio chunk into a melspectrogram
        convert the mel spectrogram to decibel scale
        normalize the spectrogram
        save each of these generated mel spectrogram to the outputdir

    return saved mel spectrograms
```

To evaluate model performance, a 80 / 20 split was used to randomly assign samples from the 45575 mel spectrograms to the training set and validation sets, respectively.

4.2.2 Model Architecture

For our experiments, we begin with two baseline models with varying architectures as shown in Figure 3 and implement different variations of hyperparameters to evaluate their effects on model performance, keeping the architectures of our models constant. For our baseline models, our input size is 48 x 128 x 1. However, as we vary the width and height dimensions of the spectrogram in subsequent versions of the model, the input size will change.

Drawing upon a number of different models previously used for bird species identification [12, 18, 19], the architecture of our models is as follows:

- **Model 1**: CNN with three 2-D convolutional layers, each of which is followed by a batch normalization layer, then a ReLU activation function, then a maximum pooling layer with a 2x2 pool size. The three convolutional layers contain 16, 32, and 32 filters, respectively. For all three convolutional layers, we use filters of size 3x3, a stride of one, and zero padding. Following the three sets of convolutional layers we implement a flattening layer followed by a fully-connected layer of size 128 with a ReLU activation and a dropout layer with a dropout rate of 50%, then a second fully-connected layer of size 11 (= number of classes in training set), with ReLU activation. Finally a softmax activation constitutes the the output layer.
- **Model 2**: CNN with four 2-D convolutional layers, each of which is followed by a batch normalization layer, ReLU activation function, and a maximum pooling layer with a 2x2 pool size. The four convolutional layers contain 16, 32, 64, 128 filters respectively. For all four convolutional layers, we use filters of size 3x3, a stride of one, and zero padding. Following the four sets of convolutional layers we implement a average pooling layer followed by two sets of fully-connected layer of size 256 with ReLU activation each followed by a dropout layer with a dropout rate of 50%. Finally a softmax activation constitutes the the output layer.

5. Results and Discussion

All results presented below use accuracy, recall and precision as the performance metrics.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Baseline models were evaluated using the model architectures detailed in section 4.2.2. The following parameters were kept constant between the two model architectures during baseline performance analysis. We used cross entropy loss with Adam optimizer, and a starting learning rate of 1e-3. A learning rate scheduler was used to update the learning rate by a factor of 0.5 if the there was no improvement after 2 epochs. Early stopping was also employed if there was no improvement in validation loss after 20 epochs. In the baseline model, 50 epochs were used with a batch size of 32. Both models were based off the starter code provided in [20] and [21].

Model	Validation set		
	Accuracy (%)	Recall(%)	Precision (%)
Model 1	73.96	64.52	89.34
Model 2	94.21	93.05	95.74

Table 1. Baseline performance of Model 1 and Model 2

Input Dim.	nfft	Validation set		
		Acc. (%)	Recall(%)	Precision (%)
48x256x1	1024	74.1	65.5	90.6
64x128x1	1024	74.9	67.1	88.4
48x128x1	2048	77.8	70.8	89.28

Table 2. Effect of spectrogram parameters on performance of Model 1

5.1. Baseline Model Performance

The input dimensions of the spectrogram in the baseline models is 48 x 128 x 1. The window of Fast Fourier Transform (FFT) was chosen to be 1024. Model 1 had a training accuracy of 75.6% at the end of 50 epochs and a validation accuracy of 74%. Model 2 had a training accuracy of 99% at the end of 50 epochs and a validation accuracy of 94%. The performance summary on validation set for both the models is shown in Table 1. From the baseline results, Model 2 is observed to perform better than Model 1. Hence few experiments such as oversampling, weighted loss, regularization, effect of spectrogram dimensions and frequency resolution were carried out to improve the performance of Model 1.

5.2. Effect of Spectrogram Parameters on Model 1

Across different studies that used CNNs for bird call classification, the input images for model training were not of uniform size. To understand the effect of spectrogram dimensions on Model 1, the hop length parameter in mel spectrogram function in librosa package is varied to produce spectrogram images of different dimensions. In this experiment, three dimensions with varying width and height size such as 48x128x1, 48x256x1, 64x128x1 are selected to understand its effect on Model 1 performance. All other hyperparameters are kept at the same level as described in section 4.2 and 5. As seen from Table 2, the change in dimensions of the spectrogram had minimal effect on the performance of Model 1. Since more than one bird may be present in any sample audio recording, increasing the frequency resolution may help the model distinguish the different bird calls. To increase the frequency resolution, the n_{fft} parameter in the mel spectrogram function was increased to 2048 while keeping the input dimensions and other parameters constant as the baseline model. As seen from Table 2, the accuracy of Model 1 increased by 4% and recall increased by 7% compared to the baseline performance.

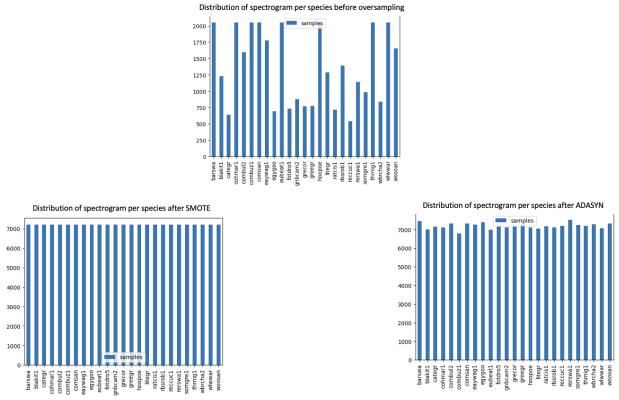

Figure 4. Distribution of spectrogram after oversampling

Technique	Batch Size	Validation set		
		Acc. (%)	Recall(%)	Preci. (%)
SMOTE	16	72.8	67.1	82.6
SMOTE	32	73.9	68.8	83.4
SMOTE	64	72.6	66.9	83.2
ADASYN	16	71.1	65.7	82.5
ADASYN	32	71.9	65.9	83.2
ADASYN	64	71.1	66.3	82.2

Table 3. Model 1 performance based on oversampling and batch size

5.3. Effect of Oversampling on Model 1

To tackle class imbalance amongst species, random oversampling for species with less representation in training data has been suggested by E Martynov et.al [22]. Two oversampling techniques such as ADASYN and SMOTE offered by the imblearn package in python were implemented for Model 1 to boost its performance. SMOTE uses k-nearest neighbour algorithm to generate synthetic data for minority classes. ADASYN is a variation of SMOTE as it generates different number of samples depending on an estimate of the local distribution of the class to be over sampled. The distribution of spectrogram per species before and after implementing SMOTE and ADASYN to the training set is shown in Figure 4. These techniques were applied to different batch sizes such as 16, 32 and 64. Irrespective of the batch size, both the oversampling techniques had minimal effect on the performance of the model as shown in Table 3. Since raw audio samples have a low signal to noise ratio, these oversampling techniques could have generated more noise than signal for the minority classes and that did not necessarily increase the efficiency of the classifier to distinguish between different bird classes.

5.4. Effect of Weighted Loss on Model Performance

In addition to the oversampling methods discussed in 5.3, we implemented `class_weight` using scikit-learn to

Model	Validation set		
	Acc. (%)	Recall(%)	Prec. (%)
Model 1	75.1	67.4	88.2
Model 2	84.6	81.2	90.1

Table 4. Model performance with weighted loss

Model	Reg.	Validation set		
		Acc. (%)	Recall(%)	Prec. (%)
4 layer CNN	None	80.6	75.1	89.9
5 layer CNN	None	83.7	79.9	89.7
2 dropout layer	None	66.9	54.1	92.1
Model 1	L1 (0.01)	74.8	63.9	90.2
Model 1	L1 (0.001)	74.3	64.9	90.3
Model 1	L1 (0.0001)	74.9	65.5	89.7
Model 1	L2 (0.01)	74.1	64.3	89.5
Model 1	L2 (0.001)	75.0	66.5	86.5
Model 1	L2 (0.0001)	75.7	67.2	89.6

Table 5. Model 1 performance based on CNN architecture

address the class imbalance present in the dataset. This feature creates a dictionary and corresponding index for each species, placing a higher penalty on misclassifications of under-represented classes, where weights are calculated as `n_samples/(n_classes * np.bincount(y))`, and `y` denotes the original class labels per sample. As shown in 4, validation accuracy and recall for Model 1 increased to 75.1% and 67.4%, respectively, while precision decreased to 88.2% compared to baseline performance. When weighted loss was implemented for Model 2, accuracy, recall, and precision decreased to 84.6%, 81.2%, and 90.1%, respectively, compared to baseline performance.

5.5. Experimentation with Model 1 architecture

To understand the effect of depth of layers on Model 1 performance, the number of convolutional layers with 128 filters, each with size 3 were increased to 4 and 5. The spectrogram parameters are similar to Model 1 baseline. The validation accuracy and recall improved by 10% and 15% compared to the baseline model by adapting Model 1 architecture with 5 convolutional layers. An additional dropout layer was also experimented with; however, it decreased the performance of Model 1. In addition, L1 and L2 regularization were included with varying degrees of penalty values ranging from 0.01 to 0.0001. For Model 1 architecture, the L2 regularizer with a penalty rate of 0.0001 yielded a 2% increase in validation accuracy and 3% increase in validation recall compared to the baseline model. The results from these techniques are summarized in Table 5.

5.6. Hyper parameter tuning:

The batch size and number of epochs were tuned for both Model 1 and Model 2 to opitimize the performance. The spectrogram features were similar to the baseline models while hyper parameter tuning. Batch sizes were varied in the range of 16,32,64 and their results are summarized in Table 6. Epochs were varied in the range of 30,50 and 70 and their results are summarized in Table 7.

Model	Epochs	Batch Size	Validation set		
			Acc. (%)	Recall(%)	Prec. (%)
Model 1	50	16	75.6	68.9	87.6
Model 1	50	64	74.1	63.4	90.1
Model 2	50	16	84.5	82.5	89.1
Model 2	50	64	84.0	80.6	90.3

Table 6. Model performance based on batch size

Model	Epochs	Batch Size	Validation set		
			Acc. (%)	Recall(%)	Prec. (%)
Model 1	30	32	74.5	65.0	90.0
Model 1	70	32	75.8	67.8	89.4
Model 2	30	32	85.0	82.7	89.5
Model 2	70	32	84.8	82.4	89.5

Table 7. Model performance based on number of epochs

5.7. Performance on Test Set

In order to avoid overestimation of accuracy, audio samples with audio rating < 4 were used for testing. The test data consists of 500 audio samples corresponding to 164 classes. From the experiments conducted, Model 2 baseline produced the highest validation accuracy and hence was used on test data. This model achieved an accuracy of 25% on the test data. Since the model was trained on only 25 classes, the robustness and generalizability of the model was low.

6. Conclusion

Deep learning techniques for acoustic monitoring of animal species, specifically CNN-based frameworks, are an invaluable tool for ecological and conservation studies. The implementation of CNN-based models for biodiversity surveying and population monitoring allows researchers to assess wildlife conservation efforts. While these methods have been used across a variety of animal species and geographic regions, we are not aware of any such models that encompass eastern African bird species. To this end, we sought to develop a CNN-based model to classify bird species in eastern Africa based on audio input of bird vocalizations.

One of the challenges in the present study was class imbalance, as many species in the dataset had three or fewer audio recordings. We experimented with a number of data augmentation and pre-processing techniques, network architectures, and hyperparameter tuning methods, to address class imbalance and to achieve high model performance. Based on our results, oversampling using SMOTE and ADASYN, as well as weighted loss, did not improve model performance, nor did increasing spectrogram height.

Since the 4-layer network outperformed the 3-layer network, it appears the addition of convolutional layers increased model performance.

Because we had limited access to GPU usage on Google Colab, we were unable to experiment with the full training set and all variations of hyperparameters. As our training data did not represent all species in the full dataset, it is possible that our smaller training set hindered model performance.

7. Future work

Further experiments will have to be conducted to understand the effects of various spectrogram features on model performance, including different values of spectrogram height (number of bins) and width (hop size and window length). Different data augmentation techniques such as increasing signal to noise ratio (SNR), pitch shifting, time stretching, amplitude scaling, introducing Gaussian noise to all samples, mixup and cut-mix augmentation should be explored, as these methods have been shown to allow further generalization of CNN models [22, 23].

References

[1] D. T. et al., "Perspectives in machine learning for wildlife conservation," *Nat. Commun.*, vol. 13, 2022. 1

[2] G. M. et al., "Method for passive acoustic monitoring of bird communities using umap and a deep neural network," *Ecological Informatics*, vol. 72, 2022. 1

[3] S. R. P.-J. Ross, D. P. O'Connell, J. L. Deichmann, C. Desjonquères, A. Gasc, J. N. Phillips, S. S. Sethi, C. M. Wood, and Z. Burivalova, "Passive acoustic monitoring provides a fresh perspective on fundamental ecological questions," *Functional Ecology*, vol. 37, no. 4, 2023. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/1365-2435.14275 1

[4] D. S. et al., "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *acoustics in Ecology and Evolution*, vol. 10, 2018. 1

[5] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks," *Detection and Classification of Acoustic Scenes and Events 2018*, 2018. 1

[6] C. Bergler, H. Schröter, R. X. Cheng, V. Barth, M. Weber, E. Nöth, H. Hofer, and A. Maier, "Orca-spot: An automatic killer whale sound detection toolkit using deep learning," *Scientific Reports*, vol. 9, 2019. 1

[7] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, "Audio based bird species identification using deep learning techniques," in *Conference and Labs of the Evaluation Forum*, 2016. 1

[8] J. LeBien, M. Zhong, M. Campos-Cerqueira, J. P. Velev, R. Dodhia, J. L. Ferres, and T. M. Aide, "A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network," *Ecological Informatics*, vol. 59, 2020. 1

[9] S.-H. W. et al., "Silic: A cross database framework for automatically extracting robust biodiversity information from soundscape recordings based on object detection and a tiny training dataset," *Ecological Informatics*, vol. 68, 2022. 1, 2

[10] J. Xie, K. Hu, M. Zhu, J. Yu, and Q. Zhu, "Investigation of different cnn-based models for improved bird sound classification," *IEEE Access*, vol. 7, 2019. 1, 2

[11] H. Goëau, H. Glotin, W.-P. Vellinga, R. Planqué, A. Rauber, and A. Joly, "Lifeclef bird identification task 2014," in *CLEF: Conference and Labs of the Evaluation Forum*, vol. CEUR Workshop Proceedings, no. 1180, Sheffield, United Kingdom, 2014. [Online]. Available: https://inria.hal.science/hal-01088829 1

[12] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000273 1, 3

[13] A. C. Lees, L. Haskell, T. Allinson, S. B. Bezeng, I. J. Burfield, L. M. Renjifo, K. V. Rosenberg, A. Viswanathan, and S. H. Butchart, "State of the world's birds," *Annual Review of Environment and Resources*, vol. 47, no. 1, 2022. [Online]. Available: https://doi.org/10.1146/annurev-environ-112420-014642 1

[14] "Ioc world bird list," https://www.worldbirdnames.org/new/ioc-lists/master-list-2/, accessed: 2023-06-01. 1

[15] H. Klinck, S. Dane, S. Kahl, and T. Denton, "Birdclef 2023," 2023. [Online]. Available: https://kaggle.com/competitions/birdclef-2023 2

[16] "Colaboratory: Frequently asked questions," https://research.google.com/colaboratory/faq.html, accessed: 2023-06-01. 2

[17] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thomé, F. Robert-Stöter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Åsen, J. Lim, A. Malins, D. Hereñú, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, and W. Pimenta, "librosa/librosa: 0.10.0.post2," Mar. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7746972 2

[18] Y. Dai, J. Yang, Y. Dong, H. Zou, M. Hu, and B. Wang, "Blind source separation-based iva-xception model for bird sound recognition in complex acoustic environments," *Electronics Letters*, vol. 57, no. 11, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ell2.12160 3

[19] M. Azeem, G. Ali, R. U. Amin, and Z. U. D. Babar, *Bird Calls Identification in Soundscape Recordings Using Deep Convolutional Neural Network*. Cham: Springer International Publishing, 2022, pp. 325–335. 3

[20] H. Klinck, S. Kahl, and T. Denton, "Birdclef 2021: Model training," 2021. [Online]. Available: https://www.kaggle.com/code/stefankahl/birdclef2021-model-training/notebook 3

[21] J. Pagirsky, "Birdclef-2021 birdcall classification," 2021. [Online]. Available: https://github.com/jeremypagirsky/birdcall_classification/tree/main 3

[22] Y. U. Eduard Martynov, "Dealing with class imbalance in bird sound classification," in *CLEF 2022: Conference and Labs of the Evaluation Forum*, vol. CEUR Workshop Proceedings, no. 3180, Bologna, Italy, 2022. [Online]. Available: https://ceur-ws.org/Vol-3180/paper-170.pdf 4, 6

[23] D. K. Arunodhayan Sampathkumar, "Tuc media computing at birdclef 2022: Strategies in identifying bird sounds in a complex acoustic environments," in *CLEF 2022: Conference and Labs of the Evaluation Forum*, vol. CEUR Workshop Proceedings, no. 3180, Bologna, Italy, 2022. [Online]. Available: https://ceur-ws.org/Vol-3180/paper-174.pdf 6