Developing an Automatic Recording Unit (ARU) with a Raspberry Pi

Aurel Agbodoyetin
AI for Science Master's programme
AIMS South Africa
Cape Town, South Africa
aurel@aims.ac.za

***Abstract*—In this study, we introduce an approach to classify bird vocalizations using a 2D Convolutional Neural Network (CNN). We improve the input audio data by adding noise and applying rolling techniques. The model's performance is assessed using standard classification metrics: accuracy, precision, recall, and F1-score. Our findings demonstrate that the suggested model attains a satisfactory classification accuracy of 85% on unseen data.**

***Index Terms*—deep learning, bird**

I. Introduction

Since the 1970s, bird populations worldwide have been experiencing significant declines. These declines are expected to continue due to climate change and changes in land management [5, 11, 22]. Monitoring avian populations through the analysis of bird calls in audio recordings is crucial for conservation efforts, scientific research, and effective ecosystem management. This method contributes to the understanding and preservation of biodiversity [7, 21]. Traditionally, this task has been done through manual surveying, often with the help of volunteers to address the challenges of scale [12, 14, 22]. However, manual observation has limitations, especially in physically challenging areas or when studying nighttime behavior. Many bird species can be easily detected through their sounds, often more so than through visual observation. Automatic species classification of birds based on their sounds has numerous potential applications in conservation, ecology, and archival purposes [15, 4, 18, 21].

In this study, we used a standard classification algorithm to analyze a vast and varied dataset of birdsong that we have collected.

II. Materials and Methods

A. Data collection

In order to develop our bird classifier, we visited Intaka Island with the purpose of capturing bird sounds and ambient noises that occur in their natural environment. Intaka Island is a 16-hectare wetland reserve located in the heart of Century City, 7km away from central Cape Town, South Africa. [9]. With its seven distinct habitats, Intaka Island is home to a remarkable variety of 120 bird species, making it a safe and easily accessible location for our project [6]. The diverse terrestrial habitat, comprised of fynbos, shrubs, and grasses, attracts birds such as the Cape Francolin, Weavers, and Cape Sparrows.

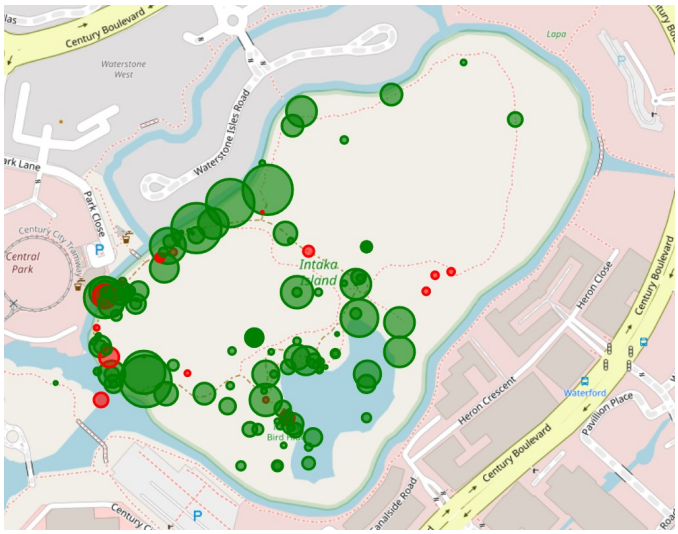

Fig. 1. Coverage of Intaka Island

We strategically positioned 13 pairs of data collectors throughout Intaka Island, as depicted in Figure 1. Each pair of data collectors was equipped with two recording kits, resulting in a total of 26 recording kits spread across the island. We assembled these recording kits using the following components:

1) Power supply
2) Microphone
3) Microphone adapter (TRS - 2 black lines)
4) Power bank cable
5) Power bank
6) Usb to 3.5mm adapter
7) MicroSD card
8) HDMI to HDMI mini
9) Pi case
10) Raspberry Pi zero
11) Usb to microUSB
12) 3-prong to 2-prong adapter

The assembled recording kit is shown in Figure 2.

Identify applicable funding agency here. If none, delete this.

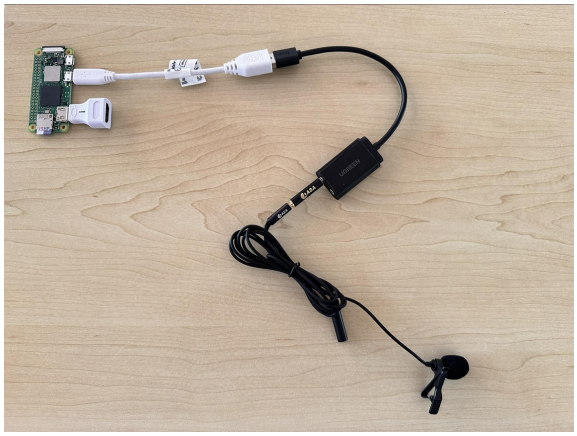

Fig. 2. Assembled recording kit

After completing our data collection, we obtained a total of 23 hours and 10 minutes of 1 channel audio data recorded at a sample rate of 44,100Hz. Our dataset at that moment consisted of many 30-second long audio clips. It is important to note that our data collectors were instructed to reboot the recording unit every 5 minutes in order to prevent any hardware or software malfunctions. It took our team one week to label the collected data using Sonic Visualizer.

B. Data pre-processing

Our data processing can be breakdown in 4 steps:

*1) **Filtering**:* We tried applying a low-pass filter at 15,000 Hz based on [8, 13, 10], as most bird vocalizations occur between 250 Hz and 8,300 Hz. This would have allow us to remove high-frequency components from the audio signals, effectively smoothing out the signal [16]. Due to ompute power limitations, we applied a low-pass filter at 9,000 Hz.

*2) **Downsampling**:* Each recording was downsampled to 22,000 Hz following the Nyquist-Shannon sampling theorem [20]. This step reduces the sampling rate of the audio signals, removing samples and reducing the amount of data representing the audio over time [19].

*3) **Creating audio segments**:* In this step, we define the duration of the audio segments to be input into the model. If the duration of the annotation is long enough, we extract multiple segments. The segments are extracted by shifting one second to the right in time. In this work, we used a segment duration of 4 seconds.

*4) **Augmentation**:* To enhance the training process and introduce variability into the dataset, we implemented data augmentation techniques on the training set. We have implemented two audio augmentation techniques, namely adding Gaussian noise and rolling.

- Noise addition: Gaussian white noise is added to each audio segment to enhance the model's resilience to real-world environmental variations. This simulates various noisy conditions that can occur naturally.
- Rolling: This technique helps the model capture temporal variations in the data, enhancing its ability to generalize to unseen data.

From this step, we obtain a 2D matrix of size `(chunk_size, 3)` by stacking the original samples with their augmented versions. Here, `chunk_size` represents the number of samples in each audio segment. We refer to this 2D matrix as *stacked amplitudes*.

*5) **Mel spectrogram**:* Each segment is transformed into a mel-scale spectrogram, which is utilized as an input image for a 2-D Convolutional Neural Network (CNN). Each dimension of the stacked amplitudes matrix is then converted into a mel spectrogram. This conversion is achieved using a Hann analysis window size of 1,024 samples with a hop size of 256 samples and 128 mel frequency bins. This process generates a spectrogram sized (128, 258) for each dimension of the stacked amplitudes. Finally, the three resulting spectrograms are combined to produce a three-channel spectrogram sized (128, 258, 3).

C. Training and testing the models

In order to train and evaluate our model effectively, we divided our dataset into training, validation, and testing sets. We randomly selected 75% of the recordings for the training set, while the remaining 25% were reserved for testing. The training dataset was then randomly divided between the training and validation sets, with a ratio of 0.8/0.2.

We trained our model for 20 epochs using a batch size of 64 spectograms and a learning rate of 0.001, with the Adam optimizer. To evaluate the performance of our model, we performed inference on the unseen data in the testing set and calculated four metrics from the confusion matrix: accuracy, recall, precision, and F1-score.

- **Accuracy** is a metric that measures how often the model correctly predicts the outcome. You can calculate accuracy by dividing the number of correct predictions by the total number of predictions [2].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- **Recall** is a metric that measures how often the model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset [2].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

- **Precision** is a metric that measures how often the model correctly predicts the positive class [2].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

- **F1-score** is a metric that evenly optimize both precision and recall at the same time [2].

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives. To assess the

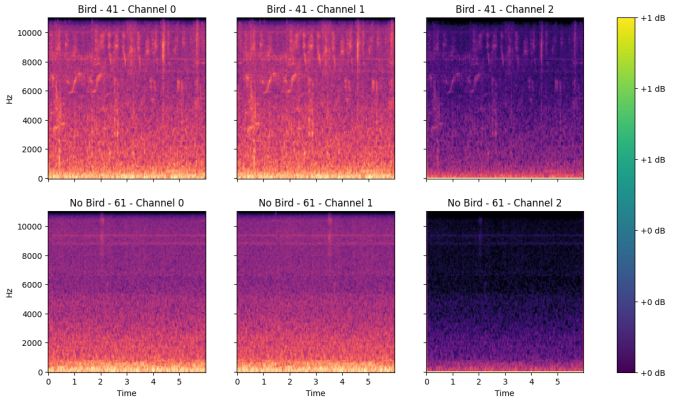

Fig. 3. Mel spectogram sample

average performance of the model, we computed the average across the ten runs for each metric.

D. Programming language, frameworks and packages

We implemented the models using Python 3 and the TensorFlow and Keras libraries [3, 1]. For audio loading and processing, we used Librosa [17], and scikit-maad [23] was used for spectrogram construction.

III. Results & Discussion

A. Preprocessing output

The three-channel spectrogram used for training is shown in Figure 3. The first row displays a presence event, while the second row shows an absence event.

B. Model architecture

The proposed model architecture is a 2D Convolutional Neural Network consisting of multiple layers of convolutional and pooling operations, followed by fully connected layers. Our proposed architecture is illustrated in Figure 4.

C. Model performance

1) Learning curves: Figure 5 displays the learning curves of our model's training accuracy and validation accuracy over 20 epochs. The training accuracy begins high and steadily increases throughout training, reaching approximately 85% by the end. This indicates effective learning in correctly classifying the training data. However, the validation accuracy initially rises but then levels off around 75% after a few epochs. This could suggest the model starting to overfit the training data. While performing well on training examples, it may struggle to generalize to new data. The training loss and validation loss show the same pattern. Thanks to our model checkpoint, we managed to restore the model from epoch 10 for inference.

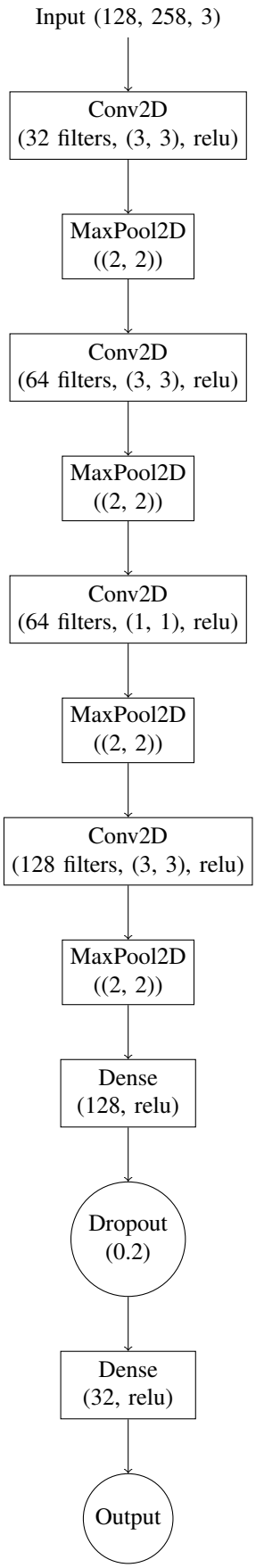

Fig. 4. Proposed Model Architecture

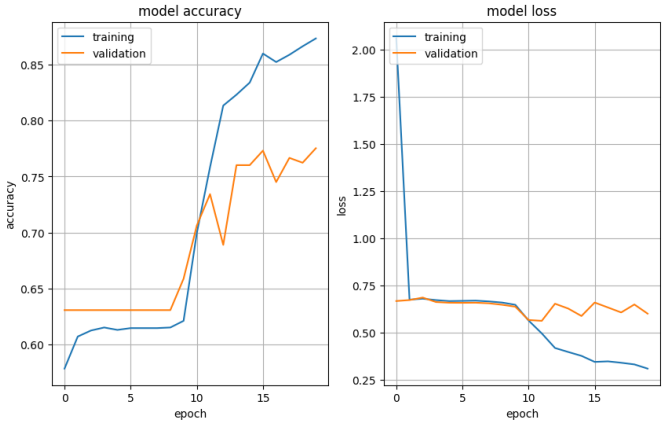

Fig. 5. Learning curves (on training and validation sets)

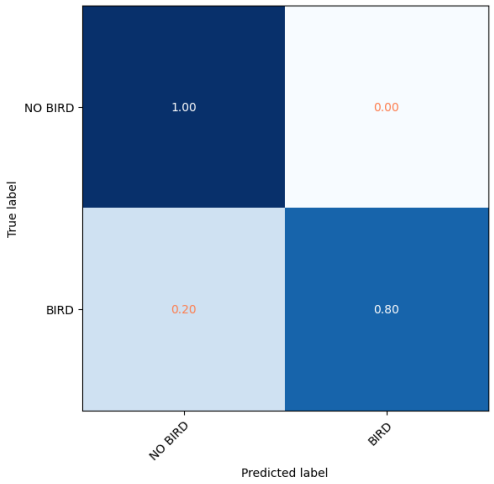

Fig. 6. Confusion matrix on test set

2) Confusion matrix & Classification report: The confusion matrix (Figure 6) shows that the model performs well in classifying audios without bird vocalizations and struggles more with images containing bird vocalizations. To gain insights into these results, we present the performance metrics in Table I.

TABLE I
CLASSIFICATION METRICS

Class	Precision	Recall	F1-score	Support
NO BIRD	0.59	1.00	0.74	32
BIRD	1.00	0.80	0.89	110
Accuracy			0.85	142
Macro avg	0.80	0.90	0.82	142
Weighted avg	0.91	0.85	0.86	142

D. GitHub repository

Our classifier can be accessed on GitHub at https://github.com/AurelAgbodoyetin/dl4e-bird-classifier, which includes the model code, data, and evaluation results.

REFERENCES

[1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467 [cs.DC].

[2] *Accuracy vs. precision vs. recall in machine learning: what's the difference?* URL: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall (visited on 02/11/2024).

[3] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[4] Andrew Digby et al. "A practical comparison of manual and autonomous methods for acoustic monitoring". en. In: *Methods in Ecology and Evolution* 4.7 (July 2013). Ed. by Luca Giuggioli, pp. 675–683. ISSN: 2041-210X, 2041-210X. DOI: 10.1111/2041-210X.12060. URL: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.12060 (visited on 02/13/2024).

[5] *Download the Report PDF*. en-US. Section: Uncategorized. Apr. 2016. URL: https://www.stateofthebirds.org/2016/state-of-the-birds-2016-pdf-download/ (visited on 02/13/2024).

[6] *Environment*. en-US. URL: https://intaka.co.za/environment/ (visited on 02/11/2024).

[7] Thomas Grill and Jan Schlüter. "Two convolutional neural networks for bird detection in audio signals". In: *2017 25th European Signal Processing Conference (EUSIPCO)*. 2017, pp. 1764–1768. DOI: 10.23919/EUSIPCO.2017.8081512.

[8] Yang Hu and Gonçalo C. Cardoso. "Are bird species that vocalize at higher frequencies preadapted to inhabit noisy urban areas?" en. In: *Behavioral Ecology* 20.6 (2009), pp. 1268–1273. ISSN: 1465-7279, 1045-2249. DOI: 10.1093/beheco/arp131. URL: https://academic.oup.com/beheco/article-lookup/doi/10.1093/beheco/arp131 (visited on 02/12/2024).

[9] *Intaka Island; Century City; Cape Town; superb birding haven; 120 species; urban wetland; Ratanga Junction (GL)*. URL: https://www.southafrica.net/gl/en/travel/article/intaka-island-century-city-urban-birding-in-cape-town (visited on 02/11/2024).

[10] Lorène Jeantet and Emmanuel Dufourq. "Improving deep learning acoustic classifiers with contextual information for wildlife monitoring". en. In: *Ecological Informatics* 77 (Nov. 2023), p. 102256. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2023.102256. URL: https://linkinghub.elsevier.com/retrieve/pii/S1574954123002856 (visited on 02/12/2024).

[11] Alison Johnston et al. "Observed and predicted effects of climate change on species abundance in protected areas". en. In: *Nature Climate Change* 3.12 (Dec. 2013), pp. 1055–1061. ISSN: 1758-678X, 1758-6798. DOI: 10.1038/nclimate2035. URL: https://www.nature.com/articles/nclimate2035 (visited on 02/13/2024).

[12] Alison Johnston et al. "Species traits explain variation in detectability of UK birds". en. In: *Bird Study* 61.3

(July 2014), pp. 340–350. ISSN: 0006-3657, 1944-6705. DOI: 10.1080/00063657.2014.941787. URL: http://www.tandfonline.com/doi/abs/10.1080/00063657.2014.941787 (visited on 02/13/2024).

[13] Stefan Kahl et al. "BirdNET: A deep learning solution for avian diversity monitoring". en. In: *Ecological Informatics* 61 (Mar. 2021), p. 101236. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2021.101236. URL: https://linkinghub.elsevier.com/retrieve/pii/S1574954121000273 (visited on 02/12/2024).

[14] Johannes Kamp et al. "Unstructured citizen science data fail to detect long-term population declines of common birds in Denmark". en. In: *Diversity and Distributions* 22.10 (Oct. 2016). Ed. by Boris Schröder, pp. 1024–1035. ISSN: 1366-9516, 1472-4642. DOI: 10.1111/ddi.12463. URL: https://onlinelibrary.wiley.com/doi/10.1111/ddi.12463 (visited on 02/13/2024).

[15] Paola Laiolo. "The emerging significance of bioacoustics in animal species conservation". en. In: *Biological Conservation* 143.7 (July 2010), pp. 1635–1645. ISSN: 00063207. DOI: 10.1016/j.biocon.2010.03.025. URL: https://linkinghub.elsevier.com/retrieve/pii/S000632071000114X (visited on 02/13/2024).

[16] *Low-Pass Filter*. en. URL: https://nl.mathworks.com/discovery/low-pass-filter.html (visited on 02/12/2024).

[17] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[18] Richard Ranft. "Natural sound archives: past, present and future". In: *Anais da Academia Brasileira de Ciências* 76.2 (June 2004), pp. 456–460. ISSN: 0001-3765. DOI: 10.1590/S0001-37652004000200041. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0001-37652004000200041&lng=en&tlng=en (visited on 02/13/2024).

[19] *Resample input at lower rate by deleting samples - Simulink*. URL: https://www.mathworks.com/help/dsp/ref/downsample.html (visited on 02/12/2024).

[20] C.E. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (Jan. 1949), pp. 10–21. DOI: 10.1109/jrproc.1949.232969. URL: https://doi.org/10.1109/jrproc.1949.232969.

[21] Dan Stowell and Mark D. Plumbley. "Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning". In: *PeerJ* 2 (July 2014), e488. ISSN: 2167-8359. DOI: 10.7717/peerj.488. URL: http://dx.doi.org/10.7717/peerj.488.

[22] Dan Stowell et al. "Automatic acoustic detection of birds through deep learning: The first Bird Audio Detection challenge". en. In: *Methods in Ecology and Evolution* 10.3 (Mar. 2019). Ed. by David Orme, pp. 368–380. ISSN: 2041-210X, 2041-210X. DOI: 10.1111/2041-210X.13103. URL: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.13103 (visited on 02/13/2024).

[23] Juan Sebastián Ulloa et al. "scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python". en. In: *Methods in Ecology and Evolution* (Sept. 2021), pp. 2041–210X.13711. ISSN: 2041-210X, 2041-210X. DOI: 10.1111/2041-210X.13711. URL: https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.13711 (visited on 10/04/2021).