Bird Vocalization Classification in Varied Acoustic Environments: A Deep Learning Approach

Abel Legese
African Institute for Mathematical Science
Cape Town, South Africa
abellegese@aims.ac.za

***Abstract*—Natural soundscapes and animal vocalizations are fascinating research subjects that provide important insights on animal behavior, populations, and ecosystems. They are investigated in the fields of ecoacoustics and bioacoustics, with a significant emphasis on signal processing and analysis. The development of more accessible digital sound recorders and significant advancements in informatics, including big data, machine learning, and signal processing, have motivated computational bioacoustics in recent years. Techniques are carried over from the larger subject of deep learning, which encompasses image and audio processing. This paper presented the process of creating an automated classifier. This includes data augmentation, preprocessing (choosing and fitting suitable neural network models), processing (choosing and fitting models), post-processing (linking model predictions to replace, or more likely facilitate, manual labelling), and processing (choosing and fitting appropriate neural network models). I trained various convolution neural networks (CNNs) on spectrograms derived from bird recordings. I found that a simple 2D CNN outperformed more complex models, achieving an accuracy of 86.76%. Data augmentation further improved the accuracy to 90.54%. However, the model struggled with non-bird sounds and misclassified some sounds as bird calls. This is due to the biased found in the two classes. The negative class is smaller in ratio. The dataset was balanced by creating a new exmple from randomly sampled negative classes. This enhance the model performance slightly. Also using spectrogram textual features and a dissimilarity frameworks helps the model better generalizes. This study highlights the potential of deep learning for bird call classification while also identifying areas..**

***Index Terms*—Ecoacoustics, Bioacoustics, Computational bioacoustics, Signal processing, Deep learning, Spectrogram, Mel frequency scale.**

I. Introduction

Birds are important for keeping nature balanced. There are lots of different kinds of birds in the world, but some are in danger of disappearing because of things like hunting and changes in their environment. When birds start to disappear, it can cause problems for other animals and plants. Birds are also good at showing us when things are changing in nature. It's important to know about the birds in an area and how many there are. But it can be hard to see birds sometimes because they're usually heard more than seen. That's why it's helpful to listen to their calls and sounds to figure out what kinds of birds are around. People have tried to figure out and group birds by listening to them, but it can be hard because there are often other noises in the background. So, experts have to spend a lot of time listening and identifying the birds. That's why it's really important to find ways to quickly and easily tell what birds are around without needing experts to do it all by hand. Traditional audio signal processing techniques and machine learning-based techniques have started playing a significant role in this respect.

Hence, a comprehensive algorithm is necessary, which can utilize the benefit of both of these approaches simultaneously. Dufourq et al., develop a deeplearning framework to classify the gibbon call form the diverse acoustic environment using passive monitoring. They revealed that a simple CNN architecture was able to predict the call with high accuracy [6]. Among the reported results, a bird species classification scheme has been suggested by [2], that was addressed using spectrogram textural features and the dissimilarity framework. According to them, this reduces the sensitivity of the model to the increase of the number of classes. Besides, Kahl et al. introduce a convolutional neural network (CNN) for large-scale bird song classification. They trained a CNN with the spectrograms from bird sounds. On the other hand, Koh et al. [13] did the same thing as in [11] with the exception that they have trained the data with predefined CNN structures like 'Resnet' [11] and 'Inception' [18]. Similarly, Incze et al. [10] also trained a pre-trained neural network, named 'MobileNet' [9] with the spectrograms obtained from the bird- calls.

Therefore, it's important to have an algorithm that can combine both of these approaches effectively. One suggested bird species classification method, as mentioned in [21], utilizes spectrogram textural features and a dissimilarity framework. This method helps reduce the sensitivity of the model to an increasing number of bird species. Additionally, Kahl et al. [11] proposed a convolutional neural network (CNN) for classifying bird songs on a large scale, training it with spectrograms of bird sounds. Similarly, Koh et al. [13] also used CNNs, specifically pre-defined structures like 'Resnet' [11] and 'Inception' [10], for the same purpose. Likewise, Incze et al. [4] trained a pre-existing neural network called 'MobileNet' [3] using spectrograms derived from bird calls.

However, Xie and colleagues [20] utilized a combination of visual and acoustic characteristics to train a Convolutional Neural Network (CNN) for classifying bird sounds. They employed the Constant Q-Transform (CQT) to convert audio data into a format suitable for input into the CNN. In terms of acoustic features, they selected spectral centroid, spectral

I would like to thank Dr Emmanuel for the support.

bandwidth, spectral contrast, spectral flatness, spectral roll-off, zero-crossing rate, signal energy, and Mel Frequency Cepstral Coefficients (MFCC). For classification using both acoustic and visual features, they compared the performance of K-Nearest Neighbors (K-NN) and Random Forest classifiers. Additionally, they analyzed bird sounds using CNN networks.

Similarly, Bold and colleagues [20] also utilized both audio and visual data to train a CNN, employing fusion strategies to integrate these cross-domain features. However, this approach, which necessitates both audio and visual data of the bird, presents challenges in practical applications. In natural settings, birds are often heard rather than seen, making automatic detection from audio alone the most viable approach, albeit still challenging.

At the same time, Lasseck et al. [2] has proposed that training a fine-tuned pre-trained deep CNN model with mel-spectrogram provides a good performance in terms of Area Under Curve (AUC) evaluation metric. Apart from all of these, according to Briggs et al. [14], the probabilistic model of short-term feature window of the audio recording and then applying Bayes Risk Minimizing Classifier provides a significant improvement to the classifier. In this case, they have used different acoustic features per frame, such as MFCCs, spectrum bandwidth, and spectrum density. They have also reported that such an approach provides similar accuracy to an SVM.

A little comprehensive real-time classifier approach was adopted by Raghuram et al. [3]. They have extracted audio frame features, such as harmonic product spectrum, energy, auto-correlative features, Tempo, pulse clarity, MFCCs, etc. Then they trained different machine learning methods like Naive Bayesian approach, SVM, Random Forest, and Neural Network. They also analyzed the classification accuracy between bird songs and bird calls. Later they combined the birds' habitat feature, bird call-type, audio recording features, and bird weight predictor to predict the bird species. Mel spectrogram was also used by Schluter et al. [15] as an input feature to the ensemble CNN and Multilayer Perceptron Models (MLPs) along with other secondary features like bird geographic location. Apart from the acoustic frequency-based spectrograms, different types of spectrograms like Gabor Transformed spectrograms have been used as the primary input feature for CNN by Heuer et al. [17]. They also analyzed the effect of different window lengths in classification performance.

In this paper I presented a set of convolution neural networks that classify the bird call from the background. In order to train a convolutional neural network (CNN)-based bird classifier, audio recordings taken from Intaka Island, Cape Town, South Africa, used as the dataset. A simple CNN classifier was successful at predicting the bird voice on diverse test set.

II. Methods

A. Data collection

The dataset utilized in this study consists of audio recordings collected from Intaka Island, Cape Town, South Africa, for training a convolutional neural network (CNN)-based bird classifier. Recordings were gathered by placing audio recorders in various habitats across the island, for almost 5 hours. Twenty-six Raspberry Pi devices equipped with audio recorders were utilized to gather bird sound data on Intaka Island. The recorders were strategically placed on trees, land, and other exposed areas to optimize sound collection from various locations.

Unfortunately, not all recording devices worked perfectly during the survey. While data was collected from 9:00 AM to 5:45 PM, there were technical issues leading to some gaps. In total, we record 4 hours of recordings from a different locations on the island. Most recordings used a sampling rate of 44,100 hz to capture sound details.

B. Data Analysis

We manually labelled a 14,006 sec recordings by inspecting spectrograms and listening to audio using Sonic Visualiser [4], and end times, the number of notes, of each observed bird and background sound. This process yielded 4823 positive and 1314 negative data. For compatibility with convolutional neural networks (CNNs) demanding fixed-length inputs, we segmented 3.9-hour recordings into 5-second intervals with 1-second overlap. Consequently, consecutive 5-second segments had 1-second time shifts. Increasing segment length allows detection of previously inaudible longer sounds. Windowing these longer segments also improves processing outcomes. Notably, the chosen window size guarantees full capture of even the longest sound within a single segment. But different downsampling rate was experimented such as 4,800hz, 9,600hz, 12,000hz, and 22,050hz were used. But given the resource constraints (RAM) the down sampling rate and the dataset size appears to be inversely proportional. At 22,050hz down sampling rate, optimal dataset size appear to be 200 recording files. Conversely for down sampling rate of 9,600hz, 350 recording files were found to be optimal.

To ensure the Nyquist rate surpassed the maximum bird call frequency, the study downsampled recordings to 4800 Hz. Each audio segment was converted into a mel-spectrogram for 2D CNN input, using a 1024-sample Hann window, 256-sample hop size, and 128 mel frequency bins. The resulting spectrogram images had dimensions of 128x38-128x173 pixels. Each down sampling rate results a different width dimensions.

C. Data Augmentation

Data augmentation is a common technique used to improve classifier performance, particularly when dealing with small training datasets and model generilization ability. In this study, two types of augmentation were used, clip distortion (1) and polarity inversion (2). Clip distortion augmentation randomly selects a percentage of points to be clipped from a uniform

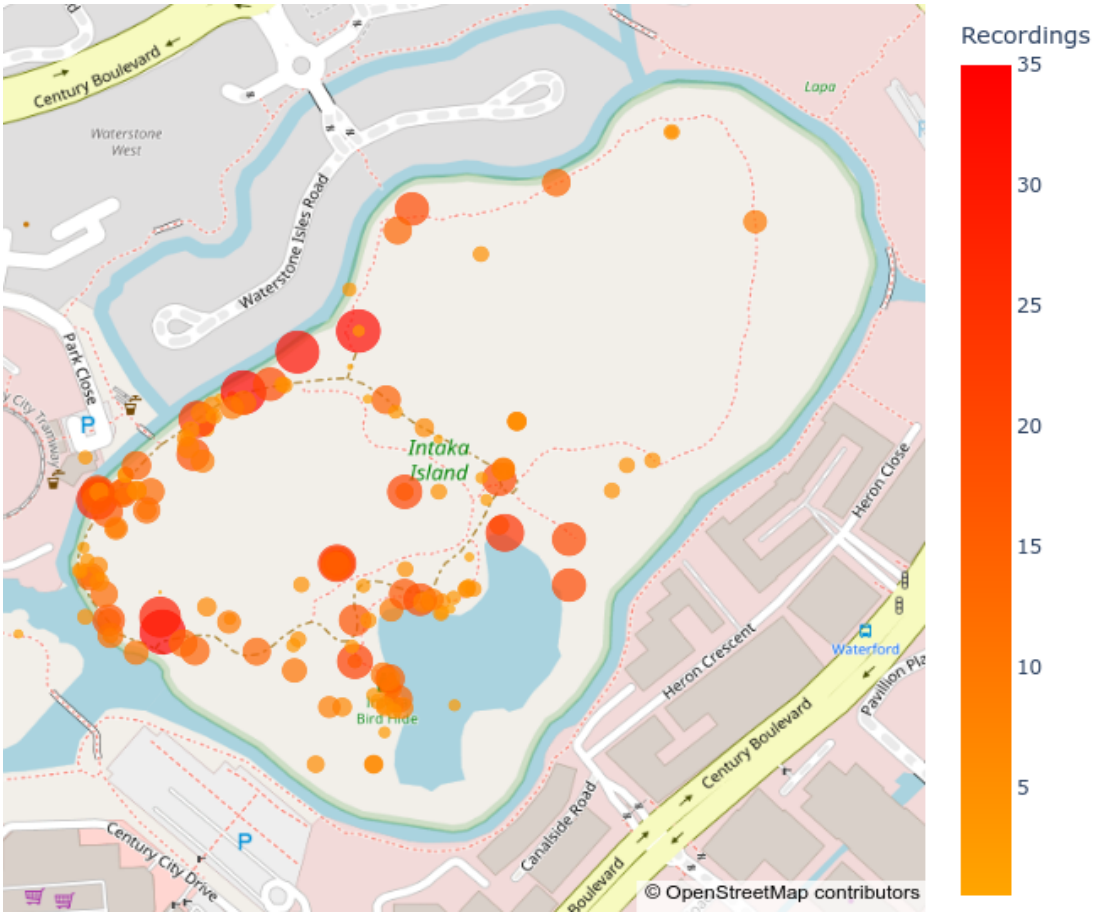


Fig. 1: A different groups distributed around different location of the Intaka Island. Twenty six people records the bird sound for 4h shifting from location to location with different recording duration and created this distribution on the map .

distribution between specified thresholds. For instance, if 30% is chosen, samples are clipped if they fall below the 15th or above the 85th percentile. This approach introduces variability in the augmentation process, enhancing model robustness. Polarity inversion method can be helpful in phase-aware machine learning model training and audio data augmentation. It is utilized for waveform differences and audio cancellation.

$$y(t) = \begin{cases} x(t) & \text{if } |x(t)| \leq A_{\text{max}} \\ A_{\text{max}} \cdot \text{sgn}(x(t)) & \text{if } |x(t)| > A_{\text{max}} \end{cases} \quad (1)$$

Where:

- A_{max} is the maximum allowable amplitude (the clipping level).
- $\text{sgn}(x(t))$ is the sign function which returns -1 for negative values of $x(t)$, 1 for positive values, and 0 for $x(t) = 0$.

Each data point in the mel spectrogram segment was augmented by generating the clip distorted version of the original spectrogram. Polarity inversion was also applied on each data point. The augmented data point were then combined in the channel dimension with the original data. This means for every original example we have three channels which helps model learns a different structure of the original image. After augmentation, a total of 18,992 segments (4,824 presence, 4,824 absence) were obtained from **578** recordings for neural network training.

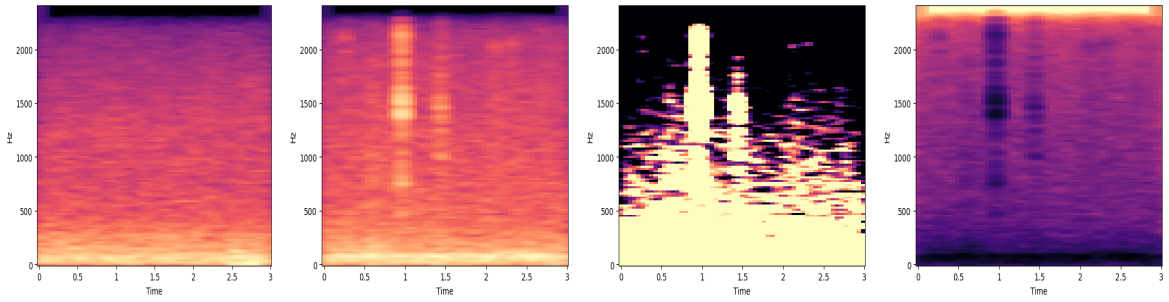
$$y(t) = -x(t) \quad (2)$$

Where:

- $x(t)$ represents the input audio signal.
- $y(t)$ represents the polarity-inverted signal.

The dataset was divided into 64% for training, 16% for validation and 20% testing, with non-augmented segments from 29 separate recordings reserved for testing. This approach ensured a diverse dataset for training and testing.

This increased diversity helps the model generalize better to unseen examples and reduces overfitting by providing more robust representations of the underlying patterns in the data. Additionally, data augmentation can also improve the model's ability to handle variations in real-world scenarios, such as changes under a inversion, noise some other transformation of the original data.



(a) Absence (b) Presence (c) Clip Augmentation (d) Polarity Augmentation

Fig. 2: The different kinds of data augmentation techniques applied on the presence data: a) absence data b) presence c) clip augmentation applied on the presence spec d) polarity augmentation applied presence spectrogram.

D. Model

The study evaluated four CNN architectures: a 2D CNN operating directly on preprocessed amplitudes of 4 second segments, and a 2D CNN utilizing spectrogram images constructed from preprocessed amplitudes with data augmentation incorporated for enhanced learning. A CNN with a large number of network parameters (e.g. EfficientNet ([19]), ResNet-10 ([8])) and MobileNetV1 [9] were also used to compare the simple 2D model. EfficientNet is a powerful CNN and efficient through special building blocks, balanced scaling, and adaptive channel focus. Each CNNs at least use up to three convolutional layers, each followed by a max pooling layer that reduces the size of the intermediate input passed to the next layer of the network.

Hyperparameters were selected based on best practices described in [7]. Each model utilized a 3x3 kernel size, suitable for processing small inputs like the 128x38-128x173 spectrograms. Batch normalization accelerated training, while Dropout regularization tackled overfitting, particularly in larger networks like ResNet-10 and EfficientNet. Each model underwent 15-40 epochs of training with Adam optimizer [12], a batch size of 32, and a learning rate of 0.001. Performance was evaluated using test set accuracy, recall, precision, and F1-score. The threshold for the classification was selected to be 0.5.

The model was implemented in Python using TensorFlow [1] with Keras [5] for the neural network and Librosa for audio processing and spectrogram generation. Training and testing occurred on a Google Colab equipped with an Nvidia T4 GPU. Parts of the code and analysis scripts were adapted from Dr. Emannuel Dufourq's project available on GitHub (https://github.com/emmanueldufourq/birdClassifier).

III. Results & Discussion

The mode was trained with two dataset size: all data points and reduced due to resource limitation. A different types of down sampling rate was experimented, such as 4,800hz, 9,600hz, 12,000hz, and 22,050hz were used. The first round I trained the model by down sampling to 9,600hz with 80%, the second one I reduced the dataset by 35% with 12,000hz, 4,800hz was also used with the full dataset and finally the model was trianed with 45% of the data with 22,050hz. Before the trained model used for testing, the audio recordings of any length was divided into overlapping 4 second segments due to different phrases for different types of bird phrases for the complete call, I took the average of the call duration which is 4 seconds. Each segment is analyzed by a trained CNN beginning at various points and producing a detection probability. This probability indicates the chances that during the next four seconds, a complete bird call will be heard. This technique makes it possible to find portions that show potential and may result in calls without having to analyze the complete tape.

The standard 2D CNN exhibited good early training behavior, with decreasing validation loss until a point around epoch 8. Training was stopped at this point to avoid potential overfitting. But the initial validation loss for EfficientNet, ResNet, and MobileNet architectures exceeded that of the 2D CNN. This likely reflects the increased complexity of these models, which may require further optimization or larger datasets to fully realize their potential. Conversely, the simpler 2D CNN appeared enough to extracting relevant features from the spectrograms, leading to its more rapid initial performance boost. The same simple model, even without data augmentation "2D CNN-No Aug", it achieves a similar score.

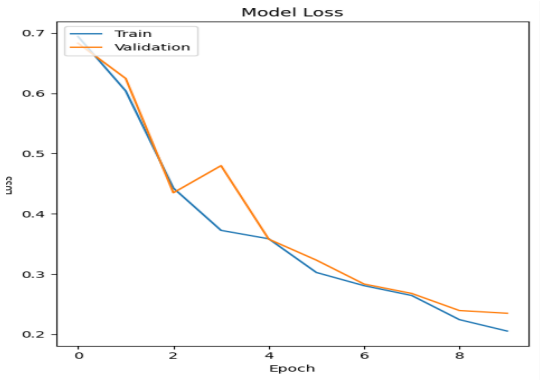


Fig. 4: Optimization progress: comparing training loss and validation loss throughout model training

The positive to negative class ratio is 4, this led to bias on

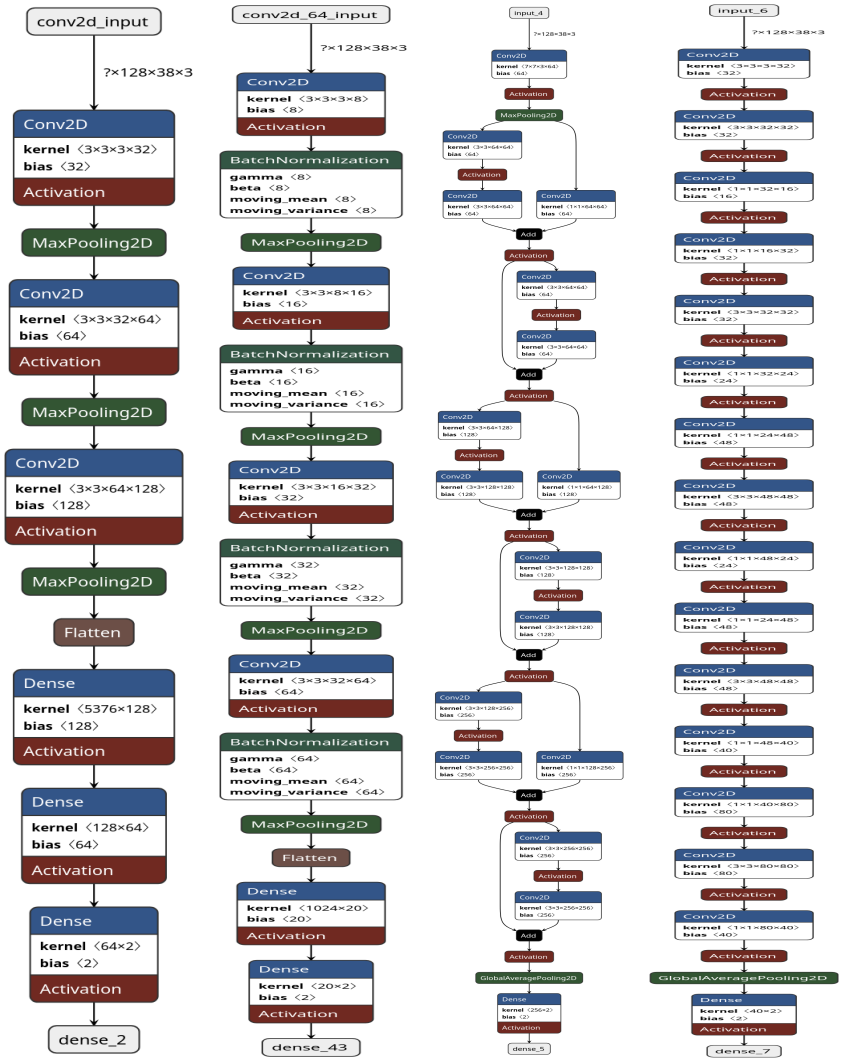


Fig. 3: Four kinds of model used on first round with 4,8000hz down sampling rate gives (128, 38, 3) dimensions a) 2D standard convolution b) The same network as (a) but with Batch Normalization and augmentation c) RestNet-10 CNN model d) Efficient Net model [Image generated by **Netron App** [16]

the final model. Due to this imbalance, the training validation loss was found to be highly unstable. To reduce the bias associated with model a new training examples were created by randomly sampling from the negative class. This created a balance between two classes. The model was trained again with the balanced dataset and shows stable loss and accuracy on the validation split.

As we can see from the Fig. 5, the model was be able to predict the presence and the absence classes given the observation above in the green line.

The provided TABLE I shows a different model performance metrics for a different model architectures. The 2D simple CNN model achieved the highest accuracy among the other models. It achieved 90.% accuracy and balanced precision and recall (90.57% and 90.54%, respectively) suggest it effectively captures relevant features from the spectrograms. This confirms previous findings that a simpler architecture is suitable for the data type with simpler complexity and sizes.

	Accuracy	F1	Precision	Recall
2D CNN	0.9053	0.8202	0.9057	0.9054
2D CNN-No Aug	0.8676	0.8010	0.7920	0.8210
EfficientNet	0.4912	0.7120	0.6932	0.6832
MobileNet	0.6121	0.7290	0.9261	0.6126
ResNet-10	0.4979	0.2534	0.4959	0.3354

TABLE I: The performance metrics applied on the four models over the 20% split test set.

While EfficientNet shows a low accuracy of 49.12%, it was not as good as the 2D CNN. This likely shows the increased complexity of this model, which might needs further optimization or larger datasets to get its a good performance

metrics. Although MobileNet has a very good efficiency, its excessively strict approach causes it to suffer from much lower accuracy (61.21%) and miss a major number of bird sounds. Lastly, with an accuracy of just 49.79%, ResNet-10 achieves the smallest of all. This might indicate that its complex architecture fall to capture a relevant features from the dataset.

Overall, the simple 2D CNN was achieved a high performance metrics in this scenario, demonstrating the value of simplicity in extracting key features from the data(spectrogram). MobileNet has shown a promise but could benefit from further fine-tuning, while EfficientNet and ResNet-10 appear less suitable for this particular dataset and application. But also a more comprehensive analysis would require additional factors like dataset size, computational resources, and specific task requirements.

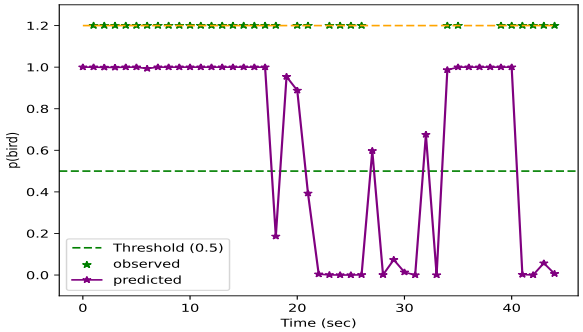


Fig. 5: The model predicts every second how likely a bird call appears within the next 45 seconds of a 9-minute test recording. A score of 0 means no chance, and 1 means presence call. The threshold set to 0.5 and the above horizontal plot shows the actual (observed) probability of the selected segments.

For non-bird class, we have a high False Negative Rate (FNR) of 89.2%, meaning nearly 9 out of 10 actual non-bird calls are missed by the model. This could imply a limitation for detecting a non-bird calls. In contrast, the False Positive Rate (FPR) is low at 8.05%. But the reason for the high FNR might be the class imbalance bias in the dataset. Even if we increase the number of example, it did help the model to detect a more diverse non-bird sounds.

For bird class, we found a different situation. The FNR drops significantly to 8.05%, indicating good accuracy in identifying actual calls. However, the FPR appears to be 89.2%, highlighting a worrying tendency to misclassify other sounds as belonging to class 2 birds.

These above performance metrics (FNR & FPR) indicates the strengths and weaknesses of our model. While achieving acceptable accuracy for one class, it struggles with sensitivity for another and generates false positives for a different type of sound. Further investigation and adjustments might be necessary to achieve a more balanced performance across all bird classes. But also collecting somewhat balanced dataset helps the mode generalizes well on both classes.

IV. Conclusion

This study explored using deep learning to classify bird calls in diverse environments. Various convolutional neural networks on spectrograms derived from bird recordings were trained. From the four types of model, a simple 2D CNN outperformed more complicated models, achieving an impressive 86.76% accuracy. Data augmentation further increased the accuracy to 90.54%. However, the model struggled with non-bird sounds, likely due to dataset imbalance, and misclassified some sounds as bird calls. This study shows the potential of deep learning for bird call classification, while highlighting areas for improvement, like non-bird sound detection and false positives for bird calls. More balanced dataset will likely remove the bias from the model that lead to better model that can predict for both classes without biases. Some research suggested that using spectrogram textual features and a dissimilarity frameworks helps the model better generalizes in the diverse bird sound dataset like we collected from the Intaka Island. For the large dataset with a more diverse vocalization, using pretrained model helps the model to learn a useful representation for different classes in the dataset.

Acknowledgment

I would like thank the AIMS Ecology team, specially Dr Emmanuel for helping me to get the data and to be a part for every important work of this research.

Code and Data Availability

You can find all the code for training and testing the neural networks, as well as doing more analyses, on GitHub here: https://github.com/Abellegese/BirdNet.git. Also, some of the sound recordings used in this study, along with labels for training and testing, are stored on Zenodo here: https://doi.org/10.5281/zenodo.10659537.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Naranchimeg Bold, Chao Zhang, and Takuya Akashi. Cross-domain deep feature combination for bird species classification with audio-visual data. *IEICE TRANSACTIONS on Information and Systems*, 102(10):2033–2042, 2019.

[3] Forrest Briggs, Raviv Raich, and Xiaoli Z Fern. Audio classification of bird species: A statistical manifold approach. In *2009 Ninth IEEE international conference on data mining*, pages 51–60. IEEE, 2009.

[4] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468, 2010.

[5] François Chollet. Keras: The python deep learning library. 2018.

[6] Emmanuel Dufourq, Ian Durbach, James P Hansford, Amanda Hoepfner, Heidi Ma, Jessica V Bryant, Christina S Stender, Wenyong Li, Zhiwei Liu, Qing Chen, et al. Automated detection of hainan gibbon calls for passive acoustic monitoring. *Remote Sensing in Ecology and Conservation*, 7(3):475–487, 2021.

[7] Ayla Gülcü and Zeki Kuş. Hyper-parameter selection in convolutional neural networks using microcanonical optimization algorithm. *IEEE Access*, 8:52528–52540, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Agnes Incze, Henrietta-Bernadett Jancsó, Zoltán Szilágyi, Attila Farkas, and Csaba Sulyok. Bird sound recognition using a convolutional neural network. In *2018 IEEE 16th international symposium on intelligent systems and informatics (SISY)*, pages 000295–000300. IEEE, 2018.

[11] Stefan Kahl, Thomas Wilhelm-Stein, Hussein Hussein, Holger Klinck, Danny Kowerko, Marc Ritter, and Maximilian Eibl. Large-scale bird sound classification using convolutional neural networks. *CLEF (working notes)*, 1866, 2017.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Chih-Yuan Koh, Jaw-Yuan Chang, Chiang-Lin Tai, Da-Yo Huang, Han-Hsing Hsieh, and Yi-Wen Liu. Bird sound classification using convolutional neural networks. In *Clef (working notes)*, 2019.

[14] Mario Lasseck. Acoustic bird detection with deep convolutional neural networks. In *DCASE*, pages 143–147, 2018.

[15] MA Raghuram, Nikhil R Chavan, Ravikiran Belur, and Shashidhar G Koolagudi. Bird classification based on their sound patterns. *International journal of speech technology*, 19:791–804, 2016.

[16] Lutz Roeder. Netron. https://netron.app/. Accessed: 2024-02-15.

[17] Jan Schlüter. Bird identification from timestamped, geotagged audio recordings. *CLEF (Working Notes)*, 2125, 2018.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[20] Jie Xie and Mingying Zhu. Handcrafted features and late fusion with deep learning for bird sound classification. *Ecological Informatics*, 52:74–81, 2019.

[21] Rafael HD Zottesso, Yandre MG Costa, Diego Bertolini, and Luiz ES Oliveira. Bird species identification using spectrogram and dissimilarity approach. *Ecological Informatics*, 48:187–197, 2018.