



AFRICAN CENTER OF EXCELLENCE IN  
DATA SCIENCE



COLLEGE OF BUSINESS & ECONOMICS

PRE-TRAINING NEURAL NETWORKS ON  
XENO-CANTO AND EBIRD FOR BIOACOUSTIC  
CLASSIFICATION MODELS

By

Mikwa Boris Tamanjong

Registration number: 220000188

A dissertation submitted in partial fulfilment of the  
requirements for the degree of Master of Data Science  
in Data Mining

University of Rwanda, College of Business and  
Economics

Supervisor: Emmanuel Dufourq, PHD

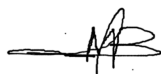
September, 2022

## Declaration

I declare that this dissertation entitled **PRE-TRAINING NEURAL NETWORKS ON XENO-CANTO AND EBIRD FOR BIOACOUSTIC CLASSIFICATION MODELS** is the result of my own work and has not been submitted for any other degree at the University of Rwanda or any other institution.

Names: MIKWA BORIS TAMANJONG

Signature

A handwritten signature in black ink, appearing to be 'MB' with a horizontal line extending to the left.

## Approval sheet

This dissertation entitled **PRE-TRAINING NEURAL NETWORKS ON XENO-CANTO AND EBIRD FOR BIOACOUSTIC CLASSIFICATION MODELS** written and submitted by **MIKWA BORIS TAMANJONG** in partial fulfilment of the requirements for the degree of Master of Science in Data Science majoring in **Data Mining** is hereby accepted and approved. The rate of plagiarism tested using Turnitin is 18% which is less than 20% accepted by the African Centre of Excellence in Data Science (ACE-DS).



---

**Supervisor**

---

**Head of Training**

## **Dedication**

I humbly dedicate this piece of work to my loving parents, Mr. Ndifim-bui Augustine Mikwa and Mrs. Mikwa Ernestine Ghranui for their endless guidance and support, to my relatives for their financial and moral supports, and my friends for their inspiring pieces of advice.

## Acknowledgement

My deepest gratitude goes to Dr.Emmanuel Dufourq for accepting to supervise this work, and for his invaluable insights, and pieces of advice throughout the execution of this project. His code <sup>1</sup> was immensely useful for the preprocessing of bird vocalizations.

Immense gratitude goes to the entire staff of the African Center of Excellence in Data Science (ACEDS) at the University of Rwanda. Your financial support, knowledge, pieces of advice, and view of life as a whole played an important part in the realisation of this dissertation.

A special thank to “The Macaulay Library at the Cornell Lab of Ornithology” for providing part of the audio recordings employed in this dissertation. I am also thankful to Xeno-canto, from which we obtained the other part of our secondary data for making its data freely available for the public. My sincere gratitude to Intaka Island Nature Reserve, in Cape Town, South Africa for granting access and allowing us to record bird vocalization.

I thank EdgeAcoustics NPO for providing support in collecting the audio data and for providing the necessary training in bioacoustic research. I also want to thank Mark Heerden for funding the AudioMoths and other audio equipment at EdgeAcoustics NPO which we used in this project.

The Annotation and the verification of the audio data were done by Dr.Emmanuel Dufourq, Aime Nshimiyimana and myself. This was a huge effort that took days of work. Thanks for the collaboration.

---

<sup>1</sup>Resource:<https://github.com/emmanueldufourq/GibbonClassifier>

I am thankful to Aime Nshimiyimana for being very resourceful. Thank you to the entire second cohort of ACEDS; you have been more than a family to me. Your love, care and support kept me going.

Finally, I can never be grateful enough to my lovely parents, Mr. Ndifimbui Augustine Mikwa and Mrs. Mikwa Ernestine Ghranui. Many thanks to my relatives for their financial or moral support. Special thanks to my loving wife, Mbambapri Sylvia Tontang for her enormous support and encouragement every step of the project and not forgetting my children, Borison-Kemuel Minuifoung and Curtis Tonui who gave me reasons to work harder. Your unconditional love was so instrumental.

## Abstract

Both traditional machine learning algorithms (linear discriminant analysis, support vector machine, decision tree, to name a few) and deep learning algorithms such as Convolutional Neural Network (CNN), Long ShortTerm Memory (LSTM), and Recurrent Neural Network (RNN) have been used in bioacoustics research in general and bird species identification in particular. However, often there is a limitation of data in bioacoustic research, including bird vocalizations. Training a deep neural network with such a small amount of data most often leads to overfitting. Many researchers have used various techniques, for instance, data augmentation and transfer learning to surpass this problem, but no research has yet been conducted on pre-training neural networks on public repositories which contain bird vocalizations, such as Xeno-canto and eBird for bioacoustic classification models. In this dissertation, we pre-trained CNNs for bioacoustic classification models using two public bird vocalization repositories (Xeno-canto and eBird) and fine-tuned them on locally collected bird audio recordings; audio recordings obtained from Intaka Island Nature Reserve, Cape Town, South Africa. First, we used bird audio vocalizations from the public repositories to pre-train three CNN models using different sample sizes. We pre-trained the three CNN models using 9000, 12000, and 15000 spectrograms (obtained by converting the audio using Fourier Transforms). Next, we trained five baseline models using different sample sizes (the entire training set, 6150, 9000, 12000, 16000, and 21000 spectrograms) from the collected data. Then, we used the same sample sizes as those employed in training the baseline models to fine-tune the pre-trained models. We used the baseline models as reference models to evaluate the performances

of the fine-tuned models. The best baseline model had a test accuracy of 91.70%, and the best-fine-tuned model achieved 91.73%. The AUC for the best baseline was 96.9% against 96.3% for the best-fine-tuned model. Three findings were observed. Firstly, the performance of the model improved when increasing the size of the training data, and secondly, the performance also improved when using the time-shift augmentation technique. Finally, the results revealed that the baseline models outperformed the fine-tuned model. The reason why the baseline models outperformed the fine-tuned model might have been because the data used in pre-training was not large enough, and a combination of CNN and RNN could produce better results. Using much larger data to pre-train the model might also improve the performance of the fine-tuned models. Despite the results, the research is the first attempt at pre-training models on publicly available bird vocalizations data that has not been investigated in the existing literature.

**Keywords:** Data augmentation; Bioacoustics; Deep learning; Pre-training.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problem Statement . . . . .	4
1.3	Rationale . . . . .	4
1.4	Objective . . . . .	5
1.4.1	Main Objective . . . . .	6
1.4.2	Specific Objectives . . . . .	6
1.5	Limitation . . . . .	7
1.6	Dissertation Outline . . . . .	7
<b>2</b>	<b>Introduction to Deep Learning</b>	<b>8</b>
2.1	Introduction to Machine Learning . . . . .	8
2.2	Introduction to Deep Learning . . . . .	10

2.2.1	Artificial Neural Networks . . . . .	10
2.2.2	Activation Functions . . . . .	11
2.2.3	Loss Functions . . . . .	13
2.2.4	Convolutional Neural Network . . . . .	13
2.2.5	Optimization . . . . .	16
2.2.6	Overfitting and Dropout . . . . .	17
2.2.7	Transfer Learning . . . . .	18
2.2.8	Evaluation Metrics . . . . .	19
<b>3</b>	<b>Literature Review</b>	<b>22</b>
3.1	Traditional Machine Learning Techniques . . . . .	22
3.2	Deep Learning Techniques . . . . .	25
<b>4</b>	<b>Data and Methodology</b>	<b>29</b>
4.1	Data Collection . . . . .	29
4.1.1	Pre-processing . . . . .	31
4.2	Pre-training CNNs for Bioacoustic Classification . . . . .	35
4.2.1	Experiment Design . . . . .	38
4.2.2	CNN Architecture . . . . .	41

<b>5</b>	<b>Results and Discussion</b>	<b>44</b>
5.1	Pre-training . . . . .	44
5.2	Baseline . . . . .	48
5.3	Fine-tuning . . . . .	50
5.4	Comparison of baseline models and fine-tuned models . . . . .	53
<b>6</b>	<b>Conclusion</b>	<b>58</b>
6.1	Future work . . . . .	60
6.2	Recommendation . . . . .	61
<b>A</b>	<b>Plagiarism Report</b>	<b>71</b>

# List of Figures

2.1	Basic architecture of an artificial neural network . . . . .	11
2.2	Architecture of a CNN . . . . .	14
2.3	Demonstration of the convolution process . . . . .	15
2.4	Illustration of max pooling using a 2x2 filter on a 4x4 input. . .	16
2.5	Typical binary classifier confusion matrix. . . . .	20
4.1	Collection of Data from Intaka Island . . . . .	31
4.2	Manual annotation of bird audio recordings . . . . .	33
4.3	Three examples of spectrograms used. . . . .	34
4.4	Training set . . . . .	36
4.5	Testing set . . . . .	36
4.6	Illustration of time shifting technique . . . . .	41
4.7	The CNN architecture used . . . . .	43

5.1	Confusion Matrices for pretrained models . . . . .	46
5.2	Visualization of the performance of all pretrained model . . . .	47
5.3	Comparison of accuracy of various baseline models . . . . .	49
5.4	Comparison of F1 score of various baseline models . . . . .	50
5.5	Comparing of baseline and fine-tuned models using AUC . . . .	54
5.6	omparison of baseline models with pre-trained and fine-tuned models. . . . .	55
5.7	learning curves . . . . .	57

# List of Tables

4.1	The distribution of secondary data . . . . .	37
5.1	Performance of various pre-trained models. . . . .	45
5.2	Average performance measures of various baseline models. . . . .	50
5.3	Average performances of various pre-trained models fine-tuned with 9000 spectrograms. . . . .	51
5.4	Average performances of various pre-trained models, fine-tuned with 12000 spectrograms. . . . .	51
5.5	Average performances of various pre-trained models, fine-tuned with 16k spectrograms. . . . .	52
5.6	Average performances of various pre-trained models, fine-tuned with 21k spectrograms. . . . .	52

# Chapter 1

## Introduction

This chapter starts with the background of the dissertation followed by the problems currently faced in bioacoustics research and ecology such as data limitation. Next, it discusses the objectives of the dissertation, and the chapter concludes with an outline of the entire dissertation.

### 1.1 Motivation

Bianco et al. (2019) defines bioacoustics as a discipline that studies how sounds is produced and perceived, especially the impact of sounds on living things and the importance of sound in communication. Information from bioacoustics is used for monitoring and conservation of ecology. This information can be used to know the dispersion, density, and migration of different species. A recent study by Dufourq et al. (2021) on the world's rarest primate, the Hainan gibbon (*Nomascus hainanus*) exemplifies this

whereby a convolutional neural network (CNN) was used, with an excellent degree of accuracy in the identification of gibbon calls in passive acoustic recordings (a noninvasive method of audio recording where a device is allowed on-site to capture sound without the presence of anyone); instead of manually listening to 8 hours of audio, only 22 minutes of human effort was needed, and the classifier could correctly identify all Hainan gibbon calls in a 72-hour recording. Conservationists are using bioacoustic data to measure the impact of their conservation efforts (Hauster, 2015). Birds are indicators of biodiversity because they provide vital ecosystem services (Priyadarshani et al., 2018; Debnath et al., 2016). Also, they are important indicators of the health of an environment (Sankupellay and Konovalov, 2018). A plethora of shallow and traditional machine learning algorithms have been used in the identification and classification of birds based on bird audio recordings (Steiner, 1981; D. Rosa et al., 2016; Acevedo et al., 2009; Ramashini et al., 2019; Debnath et al., 2016; Lasseck, 2015 ). These traditional algorithms require manual preprocessing of the audio recordings and feature engineering which are time-consuming and labor-intensive. To remedy this, deep learning algorithms such as CNN have been employed (Sprengel et al., 2016). However, deep learning requires a lot of data to train reliable models, but there is often a limited amount of data in bioacoustics because some species do not call that often, and also due to the fact that the terrain might be challenging to access. It is also hard to collect data about endangered species given that their population size is small. This data limitation has greatly hindered the application of deep learning in bioacoustics researches in that training a deep learning algorithm with small amounts of data would lead to overfitting (Xie et al., 2018). It is, therefore, necessary to augment the data or use methods



that require a little amount of data. Data augmentation techniques such as time-shifting (shifting audio to the left or right with a random second) have been used in an attempt to solve this problem (Dufourq et al., 2021; Cakir et al., 2017 ). Another method of overcoming data limitation is the use of transfer learning (Xie et al., 2018; Sankupellay and Konovalov, 2018; Tóth and Czeba, 2016).

Transfer learning is a machine learning technique used in deep learning in which parts of a network (model) that is trained on a large and potentially unrelated dataset for a given machine learning task is reused as the starting point in building a network for a new task (Bianco et al., 2019). In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another task. After the model has built, often the feed-forward layers at the end of the network are replaced with that tailored for the new task and new weights are learned for the final layer (Bianco et al., 2019). Transfer learning is often used because the model helps to extract high-level features such as edges and learned some filters which are easily transferred to a new task.

To the best of our knowledge, no pre-trained model has been built from public repositories which contain birds vocalizations. This dissertation focuses on pre-training neural networks using bird vocalizations from Xenocanto and eBirds (these are public repositories where individuals upload birds recordings for the public to use) and fine-tuning them using data collected from Intaka Island , Cape Town, South Africa, to create bioacoustic classification models. These two public repositories are chosen because they contain large volumes of bird vocalization data that are freely available to researchers.

## 1.2 Problem Statement

This research was motivated by the following reasons;

1. Bioacoustic data collection mostly results in limited data on species of interest (especially for endangered species).
2. Currently, researchers manually annotate all data; this is extremely time-consuming and unsustainable. This is not feasible if one has a large number of audio records, each lasting for long hours. The amount of human effort required to manually identify the species is enormous, and as the data set increases, it becomes limiting. (Bianco et al., 2019).
3. There is limited vocalization data for certain bird species. Some birds do not call often; in addition, it is challenging to obtain recordings of birds whose habitats are inaccessible. Furthermore, it is difficult to obtain enough data about endangered species.
4. Bioacoustic data is highly imbalanced data. That is, there is often more background environmental sounds than there are sound events for the species that one is surveying.
5. Training CNNs with imbalanced/limited data may result in overfitting.

## 1.3 Rationale

The rationales for carrying out this study are;

1. Training a CNN to automatically classify bird species based on their vocalizations will facilitate the monitoring of birds by ecologists and conservationists by reducing time and cost.
2. CNN can learn filters and does not require handcrafted filters. It will replace the manual annotation of data which is takes a of time and require intense labor. To do this manually, one has to train one's ears to become accustomed to the calls produced by the species being surveyed. This is requires enormous amount of time. Listening to long hours of recordings is time-consuming as well, and it can be prone to human errors.
3. Although transfer learning has been used in bioacoustics such as the parameter-based transfer learning by (Xie et al., 2018), no pre-trained model has been built using bird vocalizations from bird public repositories (e.g. Xeno-canto and eBird). Thus, this dissertation will produce novel results that will contribute to the body of knowledge.
4. We intend to create a public dataset for researchers so that they can use it in their research. This dataset will be published on Zenodo.

## 1.4 Objective

This section presents the main objective and the specific objectives of this dissertation.

### 1.4.1 Main Objective

This dissertation aims at determining if pre-training CNNs on existing bird audio recordings from two public repositories (Xeno-canto <sup>1</sup> and eBird <sup>2</sup>) followed by a fine-tuning step on our own collected data will provide a better classification accuracy than training the network with randomly initialized weights on the collected data only. In other words, this dissertation aims to determine the extent to which pre-training on an external dataset will improve the classification performance of a CNN.

### 1.4.2 Specific Objectives

To achieve our main objective, we formulated the following specific objectives;

1. To pre-train CNNs using a public repository.
2. To fine-tune the pre-trained models using our collected data from Intaka Island Nature Reserve Cape Town, South Africa.
3. To build CNN classification models using only the collected data.
4. To compare the performances of the classifiers mentioned in objectives 2 and 3.

---

<sup>1</sup>Xeno-canto: <https://www.Xeno-canto.org/>

<sup>2</sup>eBird: <https://ebird.org/home>

## **1.5 Limitation**

The ground truth data may have mislabeled data points due to limited domain knowledge in bird vocalizations. We could have mislabelled some audio files during data preprocessing despite our best efforts.

## **1.6 Dissertation Outline**

This dissertation contains six chapters. The first chapter covers the introduction of the dissertation; this includes the motivation, problem statement, rationale, and objective. Next, concepts used in the methodology of this dissertation are reviewed in chapter two. Chapter three discusses existing literature of bioacoustic studies. Furthermore, we present the methodology - data collection, preprocessing, and processing (model training) in chapter four. Our results are presented and discussed in Chapter five. Finally, Chapter six concludes the dissertation with some recommendations, limitations of our research, and further studies.

# Chapter 2

## Introduction to Deep Learning

It is clear from the literature that deep learning has achieved groundbreaking results in bioacoustics. This chapter introduces the reader to machine learning, and in particular, to deep learning. Section 3.1 discusses the basics of machine learning. An introduction to deep learning and related terminologies is presented in section 3.2; this includes artificial neural networks, activation functions, CNN, optimization, overfitting, regularization and dropout, transfer learning, and model evaluation metrics.

### 2.1 Introduction to Machine Learning

Designed to imitate human intelligence, machine learning is an evolving branch of computational algorithms that learn from the surrounding environment (El Naqa and Murphy, 2015). Mitchell et al. (1997) stated that “a computer program is said to learn from experience (E) with respect to

some class of tasks (T) and performance measure (P), if its performance at tasks in T, as measured by P, improves with experience E”. According to Samuel (1959), machine learning is defined as “a field of study that gives computers the ability to learn without being explicitly programmed”. There is a plethora of machine learning algorithms. Common categories of these algorithms include (Ayodele, 2010):

- Supervised learning: Functions are created that map the input to output. This involves the use of labeled data.
- Unsupervised learning: Set of inputs are modeled without labeled examples.
- Semi-supervised learning: Both labeled and unlabeled data points are used to create appropriate functions or classifiers.
- Reinforcement learning: Here, an algorithm learns how to act given observation in the real world. Each action has an effect on the environment and the environment provides feedback that guides the learning algorithm.
- Transduction: This is similar to supervised learning but does not explicitly create a function.

The history of machine learning is dated as far back as the seventeenth century (El Naqa and Murphy, 2015). According to El Naqa and Murphy (2015), Arthur Samuel from IBM first used the term “machine learning” and showed that computers could learn how to play the checker game. Rosenblatt (1958) developed one of the early neural network architectures. Since then, many breakthroughs have been made. One of them was in 1997

when Deep blue outperformed the world’s best chess player, Garry Kasparov, in a six-game match (Campbell et al., 2002). In addition, IBM Watson, in 2011, beat the two highest-ranked players in a nationally televised two-game “Jeopardy”! match (Ferrucci, 2012).

Machine learning has been used in many fields and has achieved great performances. This includes and not limited to: computer vision (Mochida et al., 2019), spacecraft engineering (D’Angelo et al., 2017), finance (Heaton et al., 2017), entertainment (Gee, 2009), ecology (Christin et al., 2019), computational biology (Angermueller et al., 2016), and biomedical and medical applications (El Naqa and Murphy, 2015).

## 2.2 Introduction to Deep Learning

Deep learning is a subset of machine learning; the term deep learning or deep neural network refers to artificial neural networks with multilayers. It has achieved groundbreaking performance in many fields such as computer vision (Voulodimos et al., 2018), speech recognition (Zhang et al., 2018), and natural language processing (Young et al., 2018), to name a few. This section introduces the reader to deep learning and related concepts as applied in this dissertation.

### 2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are the interconnection of computing units called artificial neurons (Brauer, 2018). A neuron is a node in a neu-



ral network. ANNs were inspired by the network of neurons in the mammalian cortex (Sharma, 2017). An ANN is made up of one or more layers, and each layer is made up of several interconnected neurons which have activation functions attached to them. Data enter into the network via the first layer (the input layer), then move to other layers (hidden layer) and finally, the output is obtained in the output layer. A basic architecture of ANN is depicted in figure 2.1.

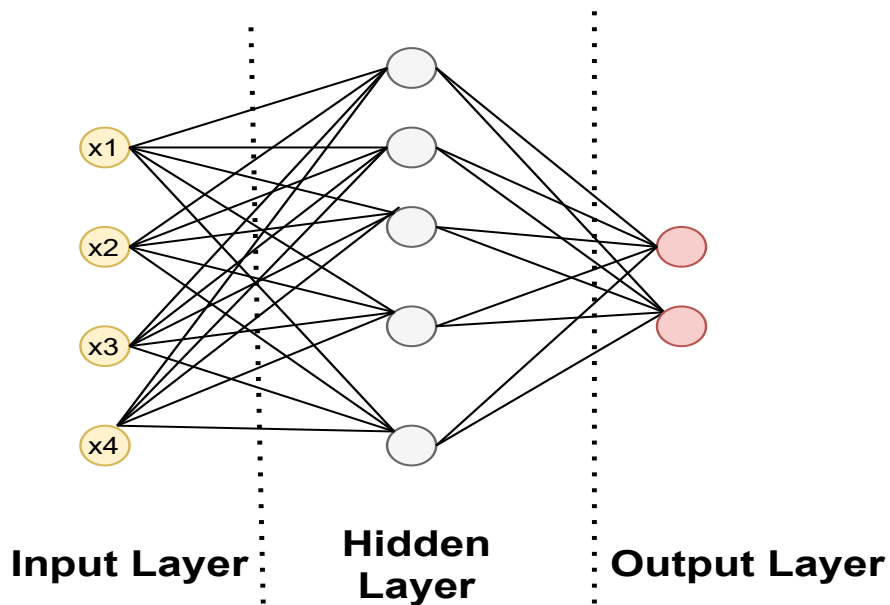


Figure 2.1: Basic architecture of an artificial neural network with one hidden layer. The input layer has four neurons, the hidden layer has five neurons, and the output layer contains two neurons.

### 2.2.2 Activation Functions

Activation functions are special functions employed in artificial neural networks to transform an input signal into an output signal (Sharma, 2017).

They transform the data past through them and produce an output. The output is then fed as input to the next layer. Activation functions play an important role in the training of a deep learning model, and the choice of an activation function depends on the task at hand. There are many activation functions such as logistic (sigmoid), softmax (Goodfellow et al., 2016), hyperbolic tangent (tanh), rectifier linear unit (ReLU)(Wang et al., 2020), gaussian error linear unit (GELU) (Hendrycks and Gimpel, 2016), to name but a few. The ReLU function is one of the most used activation functions for hidden layers. It surpasses the vanishing gradient problem (Roodschild et al., 2020) faced by the sigmoid activation function and the hyperbolic tangent activation function. Softmax activation is used in the output of multi-class classification, and it is a generalization of the sigmoid function (Goodfellow et al., 2016). The mathematical expression for the ReLU and softmax functions are shown in equation 2.1 and equation 2.2, respectively.

$$relu(x) = \max(0, x), \forall x \in \mathbf{R}. \quad (2.1)$$

$$Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (2.2)$$

Where  $z$  is a vector of input to softmax function,  $z_i$  elements of input and  $k$  is the number of classes. This dissertation uses ReLU and softmax activation functions in building bioacoustic classifiers as used by Cakir et al. (2017) and Xie et al. (2018), respectively.

### 2.2.3 Loss Functions

Neural networks are trained through an optimization process. To optimize the training process, it is important to calculate the loss; it is calculated using the loss function. This is the function being minimized (Goodfellow et al., 2016) in the optimization process. Cross-entropy and mean square error (MSE) are examples of loss functions. MSE is the most commonly-used measure (James et al., 2013) for regression problems. It is calculated by computing the average of the squared differences between the predicted and the target values. Also known as logarithmic loss, cross entropy is used for classification tasks. For both of these functions, the lower the value, the better the model. This is because lower values indicate how good the model is at predicting or classifying unseen data points. Cross entropy will be employed in this work based on the studies carried out by Cakir et al. (2017). The mathematical expression of MSE is shown in equation 2.3.

$$MSE = \left(\frac{1}{n}\right) \sum_{i=1}^n (f(y_i) - \hat{f}(x_i))^2 \quad (2.3)$$

where  $f(y_i)$  is the true observation and  $\hat{f}$  is the prediction for the  $i$ th observation ( $y_i$ ) and  $n$  is the sample size.

### 2.2.4 Convolutional Neural Network

Most of the achievements of deep learning are based on an algorithm called convolution neural networks (CNNs). It is one of the most popular deep learning algorithms. CNNs are neural networks that employ the convolution operation (instead of a fully connected layer) as one of its layers

(Ketkar, 2017). It is made up of discrete convolutions, and it was designed to mimic the connection of neurons in animal visual cortex. CNN is composed of convolutional layers, non-linearity layer, pooling layer, and fully connected layer (Albawi et al., 2017). Compared to other classification algorithms, CNNs employ relatively little pre-processing which implies that prior knowledge and human intervention in feature extraction are not a problem. The subsequent paragraphs introduce the reader to the main components of a CNN. Figure 2.2 shows a typical structure of a CNN.

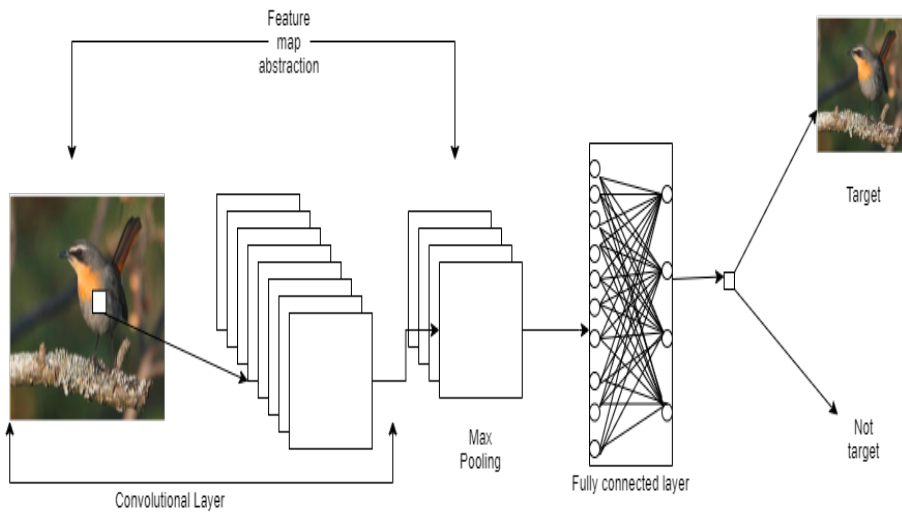


Figure 2.2: Architecture of a CNN showing the convolutional layer, pooling layer and the fully connected layer.

### Convolutional Layer

A convolutional layer is the first layer in a CNN. This layer is concerned with feature abstraction. Its input is an image. The image becomes ab-

strated into a feature map (activation map) after going through the layer as shown in figure 2.3.

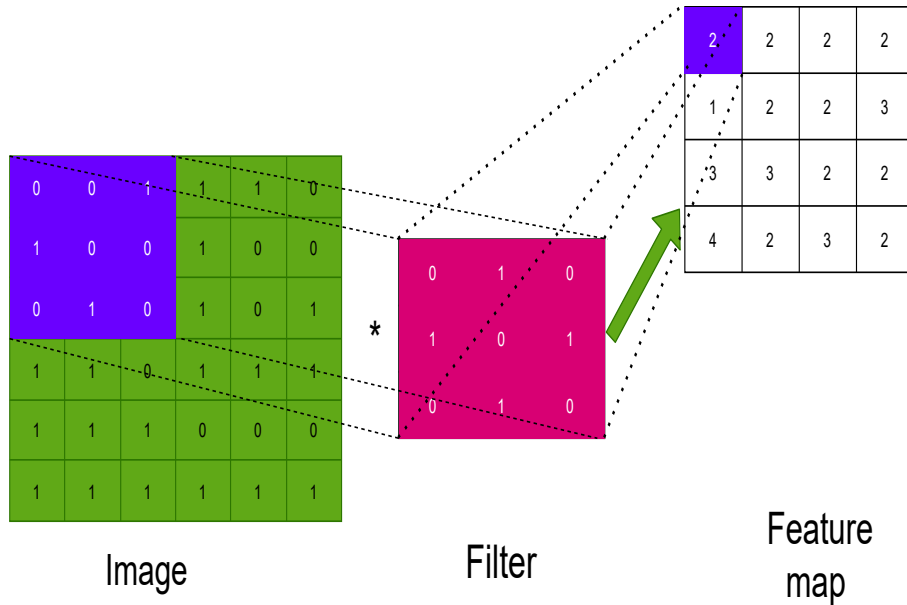


Figure 2.3: Demonstration of the convolution process using an imaginary image of size 6x6 and 3x3 filter to obtain a 4x4 output.

### Pooling Layer

This is the layer added after the convolutional layer and makes the model invariant to small changes in the input (Goodfellow et al., 2016). It is used to downsample the feature map obtained from the convolutional layer. A nonlinear activation function, in most cases, the ReLU function is applied before the pooling layer is added. Two kinds of pooling operations exist, namely: average pooling and max pooling. Max pooling computes the maximum value for each patch (a subsection of feature map that a kernel/filter processes at a time) of the feature map, and average pooling uses

the average value for each patch of the feature map. Figure 2.4 illustrates the max pooling operation. We will use max pooling in this dissertation as was used in literature by Dufourq et al. (2021).

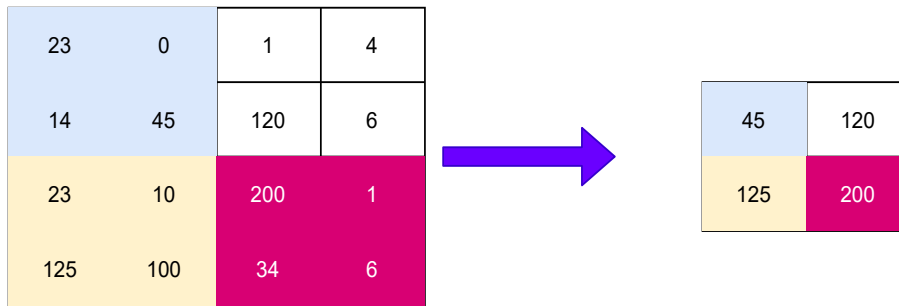


Figure 2.4: Illustration of max pooling using a 2x2 filter on a 4x4 input.

## Fully Connected Layer

Fully connected layer refers to the last few layers of a CNN. Its input is the flattened (1-dimensional array) output of the last layer of the feature extractor - convolutional layer or pooling layer. In simple terms, it is a feed-forward network. Here, every neuron in one layer is connected to every neuron in another layer. The output of the fully connected layer is fed to the softmax activation function which determines the probability of the output belonging to a specific class.

### 2.2.5 Optimization

Deep learning models are trained through an optimization process. There exists a plethora of optimization algorithms; some of which include: vanilla gradient descent, stochastic gradient descent (SGD) (Bottou, 2012), mini-

batch gradient descent (Konečn et al., 2015), Adagrad (Duchi et al., 2011), Adadelta (Zeiler, 2012), RMSprop (Hinton, 2012), adaptive moment estimation (Adam) (Kingma and Ba, 2014), to name a few. Gradient descent is the oldest optimization algorithm and is the most commonly used (Ruder, 2016). SGD is an improvement of the vanilla gradient descent algorithm; however, it depends on the manual tuning of learning rate (Soydaner, 2020) while mini-batch gradient descent uses the advantages of both SDG (SDG easily fits in the memory, and it is computationally efficient) and vanilla gradient descent. Next, Adagrad enhances the robustness (Dean et al., 2012) of SGD by adapting the learning rate. It is based on adapting the learning rate. Furthermore, Adadelta and RMSprop were developed independently almost at the same time to solve the diminishing learning rate of Adagrad optimizer (Ruder, 2016). Finally, similar to RMSprop and Adadelta, Adam optimizer employs adaptive learning rate and adds bias-correction and momentum to RMSprop. It combines the benefit of Adagrad and RMSprop (Soydaner, 2020). We will use the Adam optimization algorithm in this work. Our decision was guided by literature (Dufourq et al., 2021; Xie et al., 2018; Cakir et al., 2017).

## 2.2.6 Overfitting and Dropout

Overfitting occurs when a machine learning model does well on the data set used in training, but it does not generalize well to new examples that were not in the training data set (NarasingaRao et al., 2018). Overfitting is a major problem in deep learning. Some of the methods of preventing overfitting includes:

1. Using more data (NarasingaRao et al., 2018),
2. Data augmentation (Dufourq et al., 2021; Cakir et al., 2017),
3. Regularization: This is mostly dropout, although L1/L2 regularization are also possible (NarasingaRao et al., 2018),
4. Early stopping (Sarle, 1996),
5. Transfer learning (Xie et al., 2018; Sankupellay and Konovalov, 2018; Tóth and Czeba, 2016).

Dropout is one of the main techniques used to prevent overfitting (Srivastava et al., 2014). The main idea here is to randomly drop neurons, along with their connections from the neural network during training (Srivastava et al., 2014; NarasingaRao et al., 2018). All of the aforementioned overfitting prevention/reduction techniques have been successfully used in literature, and hence, we will employ them in this work (Xie et al., 2018; Cakir et al., 2017).

### 2.2.7 Transfer Learning

Torrey and Shavlik (2010) define transfer learning as the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. Transfer learning is a machine learning technique used in deep learning in which parts of a network (model) that is trained on a large and potentially unrelated dataset for a given machine learning task is reused as the starting point in building a network for a new task (Bianco et al., 2019). In transfer learning, a machine exploits



the knowledge gained from a previous task to improve generalization about another. After the model has been built, often the feed-forward layers at the end of the network are replaced with that tailored for the new task and new weights are learned for the final layer (Bianco et al., 2019). This technique is efficient in cases where there are data limitations (Pan and Yang, 2009; Tan et al., 2018) such as identification of bird species using bird audio recording (Xie et al., 2018; Sankupellay and Konovalov, 2018; Tóth and Czeba, 2016)

### 2.2.8 Evaluation Metrics

Evaluation metrics in machine learning are used to measure the performance of machine learning algorithms on a given task. Depending on the task, different evaluation metrics are used to evaluate a machine learning algorithm's performance. One metric can give a better measurement on one task but does poorly on another task. There are many evaluation metrics. This includes and is not limited to confusion metric, precision, recall, F1 score, accuracy, area under the receiver operating characteristic (ROC) curve (AUC), mean average precision (MAP), sensitivity, mean square error (RMSE), mean average error (MAE), to name a few. Hence, selecting a suitable evaluation metric is important for discriminating and obtaining an optimal classification model (classifier) (Hossin and Sulaiman, 2015). Here, we focus on the evaluation metrics commonly used in bird audio classification: accuracy, precision, recall, F1 score, and AUC as used in literature (Sprenkel et al., 2016; Incze et al., 2018; Tóth and Czeba, 2016; D. Rosa et al., 2016).

A confusion matrix is used in the evaluation of a model, and it is one of the most commonly used methods to present the results obtained by a classifier (Luque et al., 2019). It is made up of rows and columns; the rows represent the actual class while the column represents the predicted classes. Figure 2.5 is an example of a confusion matrix for binary classification which can be easily extended for a multi-class classifier. The confusion matrix in this case is made up of four cells designated as true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

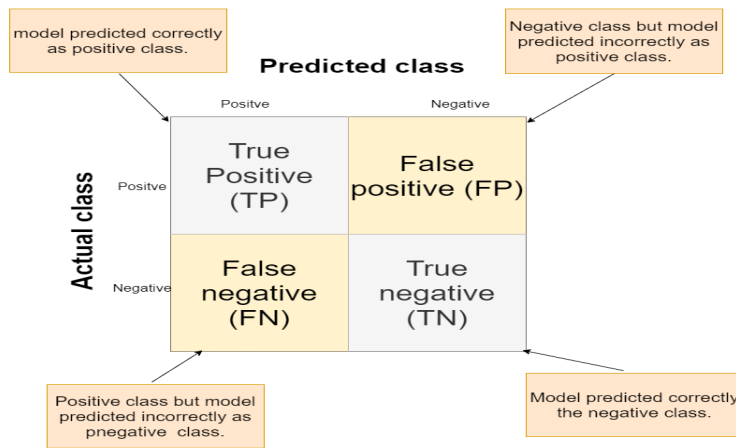


Figure 2.5: Typical binary classifier confusion matrix.

Accuracy, F1 score, precision, and recall can be defined mathematically using a confusion matrix as follows;

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.4)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.7)$$

AUC is calculated from the area under the ROC curve. ROC curve shows the true positive rate against false positive rate over various binarization threshold values (Cakir et al., 2017). This is frequently used in the classification of birds (Stowell et al., 2019; Acevedo et al. (2009); Debnath et al., 2016 )

This chapter introduced the reader to machine learning, in particular deep learning concepts. The following chapter reviews some literature in bioacoustics studies.

# Chapter 3

## Literature Review

This chapter focuses on the related studies in bioacoustics. Both shallow (traditional) machine learning techniques and deep learning algorithms have been applied in bioacoustics. The studies are grouped under two main headings: traditional (shallow) machine learning techniques and deep learning techniques.

### 3.1 Traditional Machine Learning Techniques

Many machine learning algorithms have been used in the literature and are still currently being explored in bioacoustic studies. Linear discriminant analysis (LDA) was used by Steiner (1981) to study the whistle vocalizations of five species of dolphins. D. Rosa et al. (2016) used LDA to identify and classify birds using radar data. It performed well in discriminating the presence and the absence of birds with an area under the ROC curve

(AUC) score greater than 80%. However, it performed poorly in classifying birds into various species; the AUC was less than 80%. LDA performed least among three techniques employed by Acevedo et al. (2009) to identify and classify seven frog species and three birds species. This is because of the linearity assumption (Acevedo et al., 2009) of LDA unlike the other two techniques, support vector machine (SVM) and decision tree which do not have linearity assumption. Ramashini et al. (2019) used bird sounds to classify five bird species from the Borneo Rainforest using LDA, and nearest centroid (NC) classifier with an average accuracy of 96%. Their result outperformed other techniques such as PCA/SVM (average accuracy=92%) and PCA/KNN (average accuracy=88%) using the Xeno-canto data repository (Ramashini et al., 2019); they used 50 bird calls, 10 bird calls for each species for training. Since the difference in average accuracy is small, a better comparison would have been to compare LDA/NC with PCA/NC and PCA/NC because the difference in performance would have been caused by the NC classifier and not actually by the LDA dimension reduction technique.

Debnath et al. (2016) used three classifiers (random decision tree, SVM, and extra tree regressor) in the identification of bird species using the BIOTOPE society dataset; this dataset contains 87 sound classes. Among the three different classifiers, the random decision tree method had the best performance with 96% as AUC and the worst performer was SVM with 51%. Similarly, Kampichler et al. (2010) obtained the best result using random forest and classification tree among other algorithms in the classification of ocellated turkey (*Meleagris ocellata*). However, SVM outperformed decision tree and LDA in a study carried out by Acevedo et al. (2009). SVM classifier performed as well as reference method in the automatic recognition of

bird species in a study carried out by Fagerlund (2007).

Leng et al. (2014) built an ensemble of weak classifiers (extra trees classifier, random forest classifier, KNeighbors classifier, logistic regression, SGD classifier, AdaBoost classifier, gradient boosting classifier, SVC, GaussianNB, BernoulliNB, LDA) that performed better than individual classifiers. Their result showed that a combination of weak classifiers outperformed individual classifiers in discriminating 501 bird species in the 2014 BirdCLEFF competition (Leng et al., 2014) using both audio and meta-data of bird recordings. The meta-data included: latitude, longitude, elevation, year, month, month and day, time, author of the bird recordings. The use of an ensemble of these classifiers reduced the time complexity and computational complexity. It is no doubt they would have obtained better performance if they had used strong learners but this would be at the expense of time and limited resources. The winning solution of the Neural Information Scaled for Bioacoustics (NIPS4B) 2013 (Lake Tahoe, 2013) competition used randomised decision tree classifier to classify 87 bird sound class (Lasseck, 2013). The solution involved preprocessing the audio recordings, transforming the audio to spectrograms, segmenting them, then feature extraction. The preprocessing could be minimized by using deep learning algorithms such as CNN. Lasseck (2015) improved the method in (Lasseck, 2013) by using decision tree-based feature selection and bagging to provide the basis for the winning solution to the LifeCLEF 2015 bird identification task.

Other machine learning techniques used in the bird species identification and classification include the hidden Markov model (Trifa et al., 2008), template-based methods such as time-domain matched filter (Bianco et al.,

2019) to name a few. The application of the aforementioned algorithms in bioacoustics research in general and bird species classification, in particular, require a lot of feature engineering and human input (Thomas et al., 2019). This is more costly, time-consuming, and unsustainable. Recently, researchers have resorted to alternative methods to surpass this, leveraging the deep learning techniques that have been successful in image classification (Dufourq et al., 2021).

## 3.2 Deep Learning Techniques

Deep learning has produced good results in several application areas in bioacoustic research in general and bird species identification and classification in particular. Here, we discuss some of the deep learning algorithms employed in bioacoustic research. CNN was used by Xu et al. (2017) for the detection of whales. Using Cornell University whale detection data <sup>1</sup>, they obtained AUC performance of 0.985.

Thomas et al. (2019) employed a CNN to build a detection and classification system which was able to detect and classify three species of whales, non-biological sources of noise, and ambient noise. They also proposed a new representation of acoustic signals based on spectrogram representation. They employed ResNet-50 (He et al., 2016) and VGG-19 (Simonyan and Zisserman, 2014) with batch normalization and concluded that the novel representation of acoustic signals (3 channels) improved the performance of the classifier system.

---

<sup>1</sup>Cornell University whale detection data: <https://www.kaggle.com/c/whale-detection-challenge/data>

The off-the-shelf ImageNet pre-trained ResNet-50 CNN architecture that leverages the residual learning to solved the degrading accuracy problem (the degradation problem is observed when training deep neural networks; as the network get deeper, accuracy gets saturated) in CNN architecture was used to achieve 60% to 72% accuracy of birds calls recognition using a subset of Xeno-canto dataset in the BirdCLEF 2016 <sup>2</sup> and 2017 <sup>3</sup> challenges (Sankupellay and Konovalov, 2018). In addition, the results of the work done by Tóth and Czeba (2016) in the BirdCLEF 2016 challenge show that the deep learning-based approach is well suitable for bird species classification, but fine-tuning is necessary to reach better accuracy. For instance, separating time and frequency in the CNN feature learning part and applying recurrent architectures, such as Long ShortTerm Memory (LSTM) (Hochreiter and Schmidhuber, 1997). Furthermore, the winners of the BirdCLEF 2016 Recognition Challenge, Sprengel et al. (2016) employed CNN based architectures in the classification of 999 bird species using bird audio recordings; their network architecture achieved a mean average precision score of 0.686 when predicting the main species of each sound file and scores 0.555 when background species are used as additional prediction targets.

Cakir et al. (2017) used a combination of CNN and recurrent neural network (RNN) to form the convolutional recurrent neural network (CRNN). The CRNN was used to obtain an 88.5% AUC score on the unseen evaluation data outperforming CNN on the same. The freefield1010 dataset (Stowell and Plumbley, 2013) was used for the model development, and the Chernobyl dataset was used for evaluation. This performance placed their

---

<sup>2</sup>BirdCLEF: <https://www.imageclef.org/lifeclef/2016/bird>

<sup>3</sup>BirdCLEF: <https://www.imageclef.org/lifeclef/2017/bird>



work at the second position in an automatic bird audio detection challenge.

A recent study carried out by Dufourq et al. (2021) used a CNN-based automated classifier to monitor Hainan gibbon (*Nomascus hainanus*). Like the aforementioned studies, CNN performed well in detecting gibbon calls. To solve the problem of insufficient data in bioacoustics, they used different techniques of data augmentation.

Xie et al. (2018) developed an automated bird species identification system based on multi-channel CNN. They employed the VGG-16 model (pre-trained on ImageNet) to surpass the limited training data problem in bioacoustics; the performance of the model was improved using result fusion mode. Incze et al. (2018) fine-tuned an image-based neural network (MobileNet (Howard et al., 2017) a pre-trained CNN model) using bird audio data from Xeno-canto to create a system that was able to recognize bird calls. They used the visual representation of audio (spectrograms) of different color maps; their results suggest that RGB spectrograms are more effective than their linear black and white counterparts, probably because the lower layers of MobileNet were trained on colored images. However, they could only obtain reasonable accuracies with binary classification (2 classes) and suggested that using bigger and more robust pre-trained CNN models such as ResNet (Limonova et al., 2021) might improve the accuracy of the model.

It is therefore evident to conclude that several shallow and traditional machine learning algorithms have been employed in bioacoustic research. Some of them have achieved good performances. However, they require intensive feature engineering and manual preprocessing; this is time-consuming and costly. In an attempt to surpass these problems and to step up the perfor-

mance of classification models, several studies have used deep learning, but this again faced some challenges due to data limitation resulting in overfitting. Regularization, data augmentation, and transfer learning have been used to surpass this problem. Nonetheless, we did not find any work that used a pre-trained model built from public repositories which contain bird vocalizations as a means to address the issue of data scarcity.

This dissertation focuses on pre-training neural networks using Xeno-canto and eBirds and fine-tuning them using our collected data (audio recordings) to create bioacoustic classification models. The audio recordings were soundscape data which contained calls of the Cape robin-chat (*Cossypha caffra*), and pin-tailed whydah (*Vidua macroura*) from the Intaka Island Nature Reserve, Cape Town, South Africa. Based on the literature, it is clear that deep learning is the state-of-the-art method of building classifiers for bioacoustic problems in general and bird species identification in particular. This is also the case for speech recognition task (Deng et al., 2013; Zhang et al., 2018). Given the findings presented in existing literature, We will discuss data collection and methodology in the next chapter.

# Chapter 4

## Data and Methodology

This chapter discusses the sources of data and the methodology used in this dissertation. Section 4.1 presents the data sources and the pre-processing of the data. We discuss models training and testing in section 4.2.

### 4.1 Data Collection

Both primary data and secondary data were used. Our primary data was collected using an AudioMoth (Hill et al., 2019), a passive acoustic recorder. The sampling rate was set at 48 kHz and the data was collected from 5 a.m. to 10 p.m. every day for two weeks in the Intaka Island Nature Reserve in Cape Town, South Africa. Figure 4.1 shows the site where our data was collected. The red dot in 4.1a indicates the location of the AudioMoth used to record bird vocalizations, and figure 4.1b depicts the AudioMoth. This site was chosen for the study because of the high availabil-

ity of the birds of interest there. In this project, we focused on the pin-tailed whydah (*Vidua macroura*) and Cape robin-chat (*Cossypha caffra*). Our choice of species was guided by the fact that these birds are very common and vocalize quite a lot.

The AudioMoth was attached at a height of 1.5 meters above the ground as seen in figure 4.1b. This was done in a similar way to (Darras et al., 2018; Stowell et al., 2019; Dufourq et al., 2021). A sampling rate of 48 kHz was used because it allowed us to record a wider range of frequencies. This included the sound recording of other bird species that could be of interest to other researchers since one of our objectives was to publish our dataset. Our species of interest, pin-tailed whydah and Cape robin-chat call around 3-8 kHz and 2-3 kHz, respectively. Using the Nyquist theorem (Landau, 1967), a sampling rate of 16 (8\*2) kHz suffices for both species. 8 kHz is the maximum frequency, and it should be doubled to avoid artifacts. Choosing a much less sampling rate would lead to aliasing, an effect that causes different signals to become indistinguishable when sampled.

Secondary data was obtained from bird vocalization public repositories, Xeno-canto, and eBirds. Xeno-canto is a website for sharing recordings of sounds of wild birds from all over the world, and eBird gathers unprecedented volumes of information on where and when birds occur at high spatial and temporal resolutions. We downloaded 60 pin-tailed whydah audio recordings and 125 audio recordings of Cape robin-chat from Xeno-canto; this was the amount of data available in this library when this study was done. The data had different qualities, categories A to E. Category A was the best quality (clear recordings with little or no background noise) while category E represented the worst quality (unclear recordings with a lot of

background noise). One hundred and forty audio recordings of pin-tailed whydah and Cape robin-chat were provided by eBird (Cornell Lab of Ornithology).



(a) Intaka Island Nature Reserve  
Cape Town, South Africa.



(b) AudioMoth hung at a height of  
1.5 meters at a sampling rate of 48  
kHz for audio recording.

Figure 4.1: Collection of Data from Intaka Island, Cape Town. The red dot in (a) shows the location of the AudioMoth used in recording bird vocalizations and (b) shows the AudioMoth.

### 4.1.1 Pre-processing

Librosa library <sup>1</sup> and Sonic Visualiser <sup>2</sup> were used to preprocess our data. The data were manually labeled using Sonic Visualiser (an application for viewing and analyzing the contents of music audio files). It is an open-source software that is used in visualizing, analyzing, and annotating sound files. We chose it because of our previous experience with the software.

---

<sup>1</sup>librosa library: <https://librosa.org/>

<sup>2</sup>Sonic Visualiser: <https://www.sonicvisualiser.org/>

The audio files of bird vocalizations are a sequence of amplitudes sampled at a specific sampling rate. The sampling rate is defined as defines the number of samples per second taken from an analog (continuous) signal to create a digital signal. The sampling rate of 48 kHz means that 48000 samples were taken every second from the analog signal during recording to make the digital signal shown in Figure 4.2 as a waveform. The calls were of varied duration, ranging from 3 seconds to 30 seconds. The detailed annotation process is explained in the next paragraph.

The annotation process consisted in deciding labels to use, and then labeling the audio files in Sonic Visualiser. We labeled the signal into three classes: pin-tailed whydah (PTW), Cape robin-chat (CRC), and noise - every other sound different from PTW and CRC was considered as noise. We used only three classes because we wanted to focus solely on two species of birds; this is because they are very common and call very often. We employed both visual and auditory techniques to label the files. First, we had to load an audio file into Sonic Visualiser; it appeared in the waveform as shown in figure 4.2. Next, a spectrogram of the 20 minutes long audio was displayed in Sonic Visualiser. Then, the audio was listened to while visually comparing the call pattern with reference calls. Once, a call was identified, a bounding box was drawn around the calls as shown at the bottom of figure 4.2. In addition, the label is written in an editor as seen in figure 4.2. A similar procedure was followed to label noise. Finally, the labeled file was exported and saved as an SVL file.

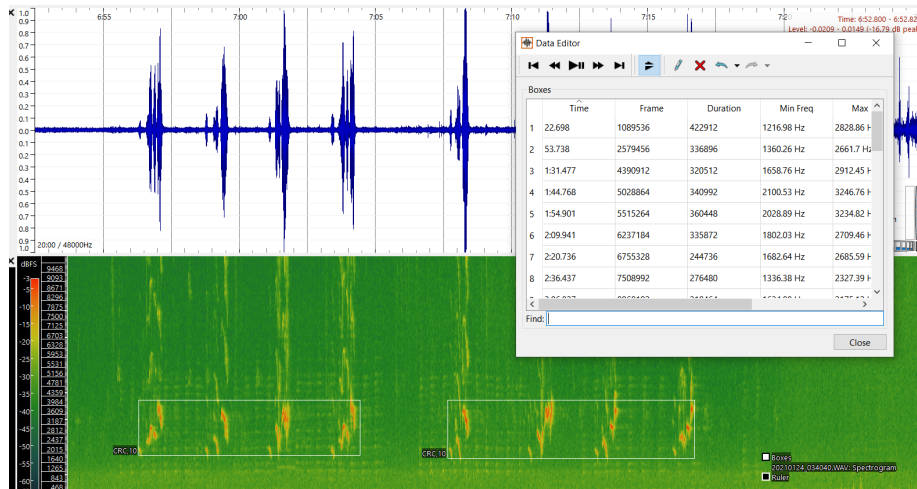
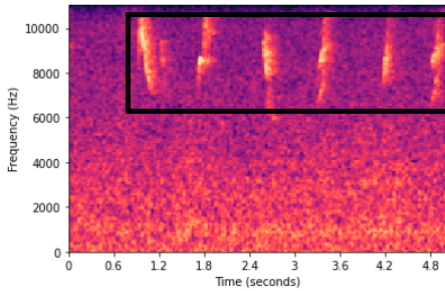


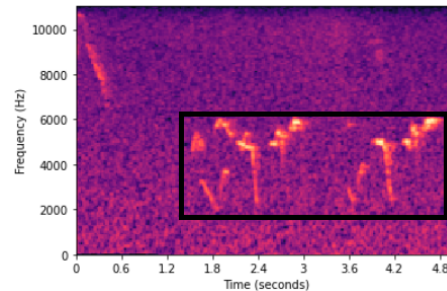
Figure 4.2: Manual annotation of bird audio recordings into three (3) classes: pin-tailed whydah, Cape robin-chat and noise. The top right portion shows the amplitude of a typical file in Sonic Visualiser. Cape robin-chat calls enclosed in boxes in the spectrogram at the bottom. To the top right is editor with all the labels.

The outcomes of the labeling process were SVL files that were further processed using the librosa python library. Librosa was used to downsample the audio file to 16 kHz. This means taking 16000 samples per second. The audio file was downsampled to reduce the computational complexity and to extract only the frequency range in which the two birds call. We used a 16 kHz sampling rate because the maximum calling frequency of the species of interest is 8 kHz; this was done in accordance with the Nyquist theorem which states that a periodic signal must be sampled at more than twice the highest frequency component of the signal. Then, various equal (CNN requires images of the same sizes) length segments were extracted with the aid of the SVL file guided by domain knowledge of the calling species- minimum and maximum call duration, and minimum and maximum frequen-

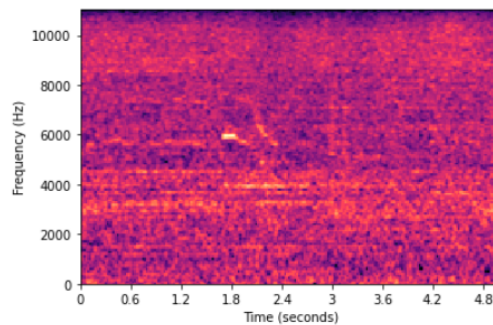
cies (Xie et al., 2018). Three-second segments were used because the shortest duration of the birds' call was 3 seconds. The segments were then converted into mel spectrogram; a spectrogram<sup>3</sup> is simply a two-dimensional visualization of a sound. Elapsed time is represented along the x-axis, frequency is represented along the y-axis, and amplitude is represented by color intensity. These are the images that were inputted to the 2D-CNN (Dufourq et al., 2021). Three examples of such spectrograms are shown in figure 4.3 with bounding boxes drawn around calls. The next section details how the CNN models were trained and tested.



(a) Spectrogram of pin-tailed whydah of size 128x216 with its calls enclosed in a rectangle.



(b) Spectrogram of Cape robin-chat of size 128 x 216 with its calls enclosed in a rectangle.



(c) Spectrogram of noise size 128 x 216.

Figure 4.3: Three examples of spectrograms used.

<sup>3</sup><http://soundbirding.org/index.php/sound-and-spectrograms/>



## 4.2 Pre-training CNNs for Bioacoustic Classification

This section details how we pre-trained CNNs for bioacoustic classification. Here, we employed 2D-CNN s was used in literature (Xie et al., 2018; Dufourq et al., 2021) to pre-train our model using spectrograms. Baseline classifiers were built using bird audio recordings obtained from the Intaka Island Nature Reserve Cape Town, South Africa. Then we pre-trained three classifiers on the bird vocalizations from the public repositories and fine-tuned them on the data collected in Cape Town. Section 4.2.1 presents our experimental design. We split the data collected into a training set and test set as shown in figure 4.4, and figure 4.5, respectively. The training set from the collected data was used in building the baseline model and for fine-tuning pre-trained models. Our code and script can be found at [github](#).

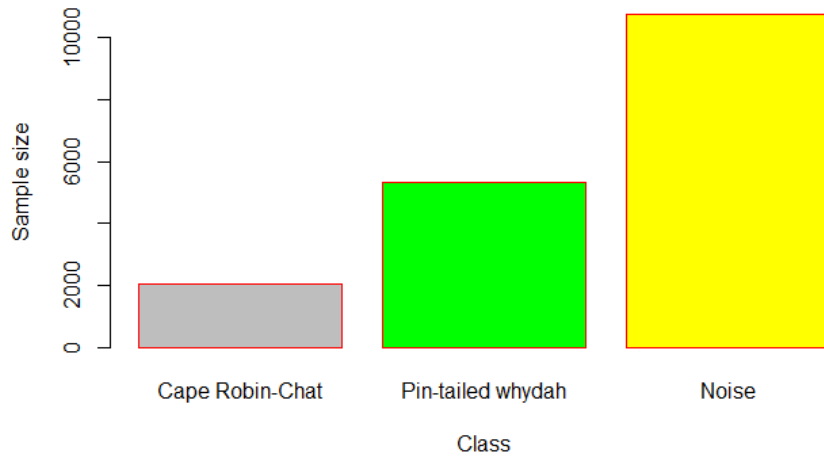


Figure 4.4: Training set made up of 18087 segments; 2050 Cape robin-chat segment, 5315 pin-tailed whydah, and 10722 Noise

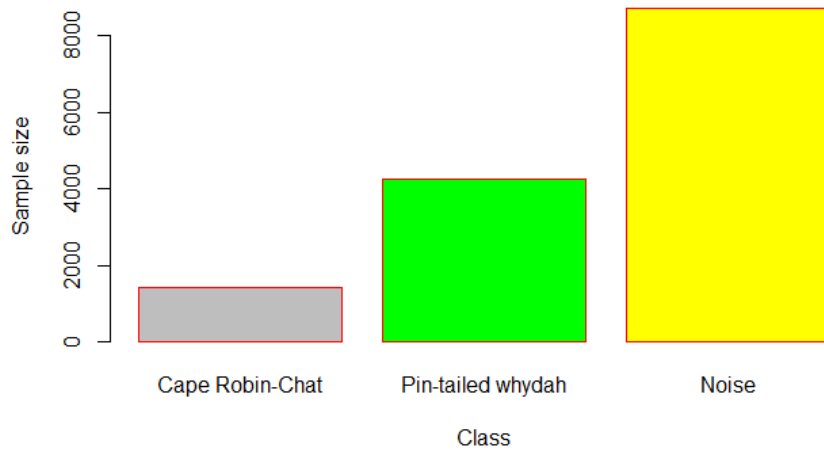


Figure 4.5: Testing set made up of 14416 segments; 1436 Cape robin-chat segment, 4264 pin-tailed whydah segment, and 8712 noise.

Table 4.1 shows the distribution data points for the various class used for pre-training our models.

Table 4.1: The distribution of secondary data for model pre-training.

Secondary dataset	
Class	number of data points
Cape robin-chat	5045
pin-tailed whydah	1980
Noise	425

### 4.2.1 Experiment Design

---

**Algorithm 1** Experiment Design

---

- 1: Split data collected from Intaka Island into training and testing. The Testing data is constant.
  - 2: Sample  $X$  (e.g. 50 or 100) amount of calls from Xeno-canto and eBird.
  - 3: Sample  $Z$  (e.g. 1000 or 2000) amount of background noise from training data obtained in step 1.
  - 4: Augment,  $X$  data obtained from step 2, to the same amount as  $Z$ . This will ensure that both  $X$  and  $Z$  represent a balanced dataset.
  - 5: Pre-train a classifier,  $C$ , on the data from steps 4 and 3. That is, on  $X$  and  $Z$  spectrograms.
  - 6: Apply model to test data obtained from Xeno-canto and eBird.
  - 7: Sample  $Y$  (e.g. 70 or 120) amount of calls from training data obtained from step 1.
  - 8: Augment,  $Y$  data obtained from step 7, to the same amount as  $Z$ . This will ensure that both  $Y$  and  $Z$  represent a balanced dataset.
  - 9: Sample  $Z$  ( $Z$ , same value from step 3) amount of background noise obtained from step 1.
  - 10: Fine-tune the classifier,  $C$  the on data obtained from steps 7 and 8. That is,  $Y$  and  $Z$  spectrograms.
  - 11: Train a randomly initialised classifier,  $I$ , on data obtained from step 7 and 8. That is,  $Y$  and  $Z$  spectrograms.
  - 12: Apply classifier,  $I$ , to test data obtained from step 1.
  - 13: Repeat steps 2 to 10 by changing the values of  $X$ ,  $Y$  and  $Z$ , and compare the results obtained by classifier  $C$  (pre-training and fine-tuning) and  $I$  (randomly initialised).
-

The data obtained from Intaka Island was split into a training set and test set, in the ratio 6:4. The test set was kept constant but different samples of the subsets of the training set were used in fine-tuning the pre-trained models to see the impact of different data sizes on the fine-tuning process. A varying number of calls, for instance, 900, 1200, etc were randomly selected from the secondary data obtained from Xeno-canto and eBird (secondary data) and used in pre-training CNN networks to measure the impact of data size on pre-training a model. To prevent background noise from affecting (not learned by the classifier) our classifier, for each sample of calls selected from the public repositories, we added a sample of background noise from the collected data to ensure class balance. The time-shifting augmentation technique was employed to ensure class balance in the secondary data set. The augmented data was then used in pre-training CNN networks. The pre-trained networks (models) were applied to the test data obtained from the secondary data set. This was repeated 6 times and the performance (F1, Test accuracy, precision, recall, and AUC) was averaged.

We provide an example to make the explanation clearer. We randomly selected  $X$  calls from Xeno-canto and eBirds and randomly selected  $Z$  background noise from the training set. The 50 calls (just for illustration) from Xeno-canto were augmented to  $Z$  calls using time-shifting to ensure class balance between the two classes (spectrograms that contain calls and those that do not). The result of this was that  $X$  plus  $Z$  examples were available for training. Next, the examples were used to pre-train a classifier. After pre-training,  $Y$  calls were randomly chosen from our collected data from Cape Town, and then augmented to the same value as  $Z$  using time-shifting augmentation technique. Again  $Z$  background noise were randomly

selected and used with the augmented data to fine-tune the pre-trained model. The augmentation steps were used to create balanced datasets.

The  $Z$  background noise samples together with the augmented data were used to train a randomly initialized (baseline) model. Finally, the test set was used to evaluate the performances of both the fine-tuned model and baseline. The procedure was repeated using different sample sizes from the training set and Xeno-canto and eBirds calls, and the performance of the models is averaged. Variable sizes of training data was used to measure the impact of the size of training data on the performance of a classifier. This process is summarised in Algorithm 1. The proposed methodology allows us to test our hypothesis which is to determine if pre-training CNNs on existing bird audio recordings from two public repositories (Xeno-canto and eBird) and fine-tuning using the audio data we collected locally will provide a better classification accuracy than training the network with randomly initialized weights.

Time shifting is illustrated in figure 4.6. Figure 4.6a is the original spectrogram while figure 4.6b and 4.6c are the time-shifted versions. They are time shifted to the right. When the spectrogram is shifted, the data is wrapped back to the left. This results in realistic synthetic data which would be a representation of true calls recorded in the environment. Figure 4.6c shows the spectrogram of figure 4.6b wrapped to the left.

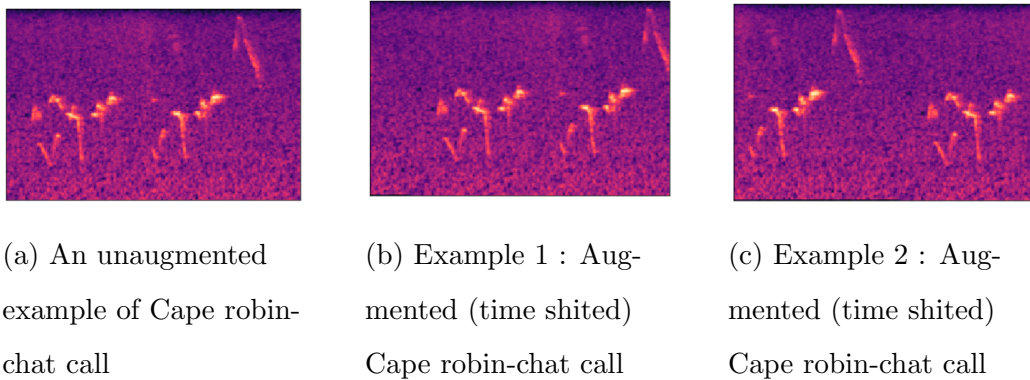


Figure 4.6: Illustration of time shifting technique using Cape robin-chat segments.

## 4.2.2 CNN Architecture

We trained our models on Google Colab Pro using the Keras API. Figure 4.7 shows the architecture that was used. It is made up of 15 layers. We used Adam optimizer as used in literature (Dufourq et al., 2021; Xie et al., 2018; Cakir et al., 2017). Also because it is computationally efficient, requires little memory space and works well with large data sets. This was chosen based on literature. The ReLU activation function was used for the input, convolutional layer, and dense layer as guided by Cakir et al. (2017)), and the softmax activation function was employed similarly as used by Xie et al. (2018). This is because this is multi-classification problem and the softmax function returns the probabilities of each class; the target class is the class with the highest probability. The input to the CNN were spectrograms of size 128 x 216 each, and shape 128x216x1.

Early stopping having a patience of 10 was employed. This was guided by the work done by Tóth and Czeba (2016). A dropout rate of 0.5 was em-

ployed; our choice was in accordance with a recent study carried out by Dufourq et al. (2021). Both early stopping and dropout were used to prevent overfitting. Many hyperparameters were experimented with and the best results were obtained using the following; 32 filters of size 3x3 in each convolutional layer, a kernel size of (2,2) in every max-pooling layer, and two dense layers of 32 and 3 hidden nodes. A learning rate of 0.0001 was used when fine-tuning the pre-trained models.



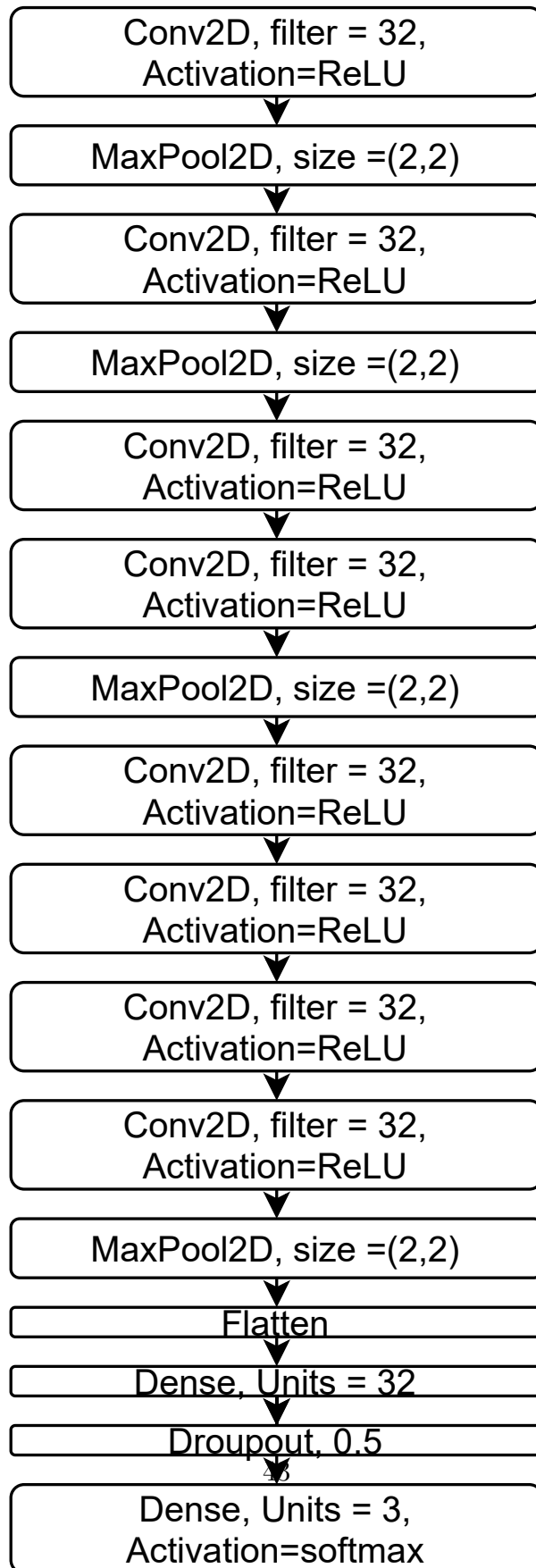


Figure 4.7: The CNN architecture used. It is made up of 15 layers (8 convolutional layers, 3 max pooling layers, 1 flatten layer, 1 dropout layer and

# Chapter 5

## Results and Discussion

In this chapter, we present and discuss the results of our findings. The results are presented following our objectives which include: 1) to pre-train CNNs using a public repository, 2) to fine-tune the pre-trained models using collected data, 3) to build CNN classification models using only the collected data, and 4) to compare the performances of the classifiers. Sections 5.1, 5.2, and 5.3 present the results for pre-training with data from Xeno-canto and eBird, baseline models using our collected data, and fine-tuning the pre-trained with collected data, respectively. Finally, section 5.4 compares the baseline models and the fine-tuned models.

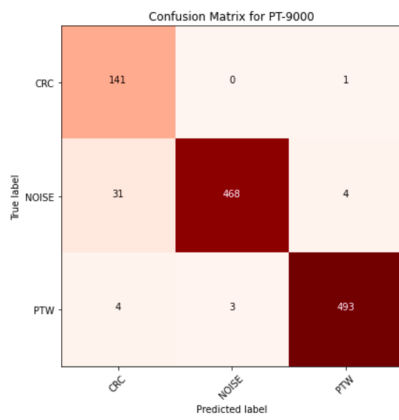
### 5.1 Pre-training

A total of three experiments were conducted. The first experiment was denoted as PT-9000. It was pre-trained using 9000 samples (spectrograms);

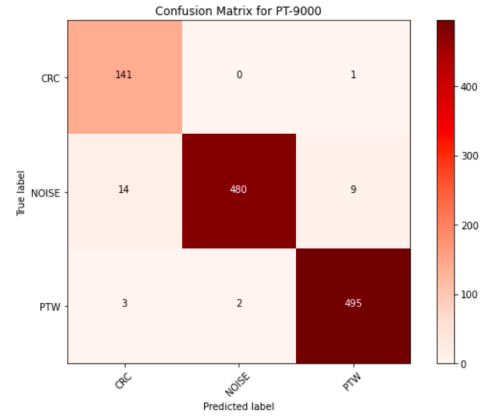
three thousand samples from each class. We denoted the second experiment as PT-12000. It was trained using 12000 samples; four thousand samples from each class. The third experiment was denoted as PT-15000. We used 15000 samples, 5000 samples from each class. Then, 1145 (142 Cape robin-chat, 503 noise, and 500 pin-tailed whydah) samples from Xeno-canto and eBird were used to check the performance of each of the models. Figure 5.1 shows the confusion matrices, table 5.1 and figure 5.2 summarises the results obtained. The results show that the larger the sample size, the higher the AUC, precision, F1 score, recall, and AUC values. PT-9000 model performed the least while the PT-15000 performed the best.

Table 5.1: Performance of various pre-trained models.

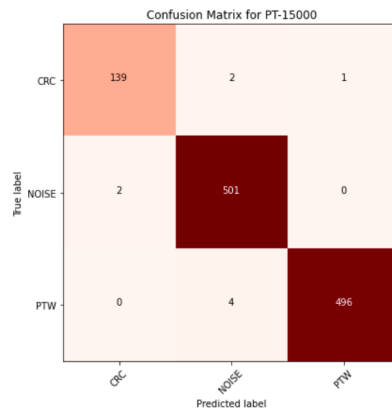
Performance Of Various Pre-trained Models			
Performance measurre	PT-9000	PT-12000	PT-15000
Precision	0.9282	0.9562	0.9907
AUC	0.9957	0.9985	0.9998
F1 Score	0.9453	0.9666	0.9898
Accuracy	0.9624	0.9747	0.9921
Recall	0.9698	0.9791	0.9890



(a) Model pretrained on 9000 spectrograms.



(b) Model pretrained on 12000 spectrograms.



(c) Model pretrained on 9000 spectrograms.

Figure 5.1: Confusion Matrices for pretrained models obtained using test data from Xeno-canto and eBird

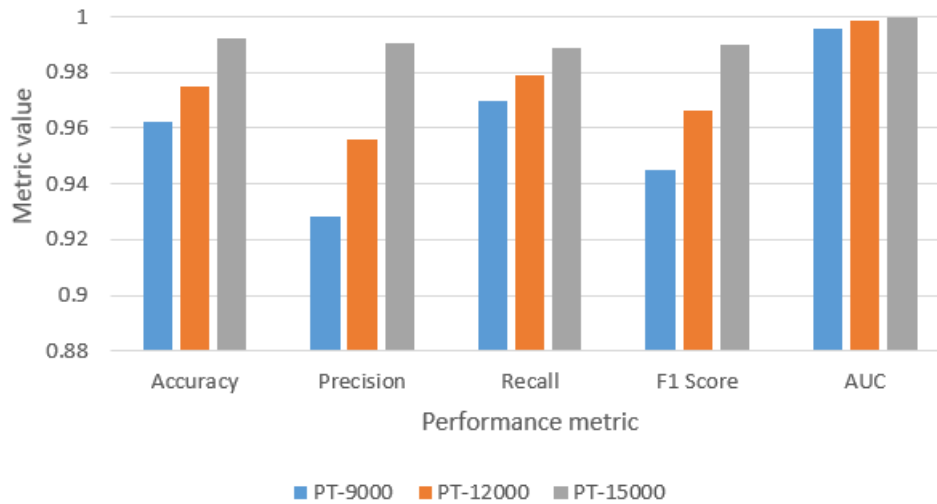


Figure 5.2: Visualization of the performance of the different pretrained models on test data obtained from Xeno-canto and eBird

The confusion matrix in figure 5.1 shows that the pre-trained model performs well in classifying various calls into the respective classes. The confusion matrices are actually pretty good for all the pre-trained models. For the PT-9000, figure 5.1a, 141 CRC were correctly classified as CRC, and 1 CRC is classified as PTW. There were 31 and 4 NOISE samples that were misclassified as CRC and PTW, respectively. Only 7 of the 500 PTW used were not correctly classified. The PT-12000 model confusion matrix in figure 5.1b indicates the PT-12000 classifier had the same performance as PT-9000 model for the CRC class. The PT-12000 model did slightly worse than the PT-9000 for the NOISE class as 43 noise were incorrectly classified. However, it performed better than PT-9000 for the PTW as only 5 PTW calls were misclassified. The PT-15000 model had the best performance for the NOISE and PTW classes because only 2 NOISE and 4 PTW samples were misclassified. These results show that we will have few false

negatives for PTW and CRC, The models also did excellently in discriminating noise from the other two classes. This means that our models will correctly determine the noise and we won't have a lot of false positives. In ecology, we record thousands of hours of data, and most often this data contains many hours of background noise with very few examples of calls. We always want to reduce false positives as much as possible and our result illustrates this.

## 5.2 Baseline

Five baseline models were built using different subsets of the training set as indicated in the methodology and evaluated using the test set. The models were denoted as M\_6150 model, M\_9000 model, M\_12000 model, M\_16000 model, and M\_21000 model. The number in the name of the models represents the size of the subset of the training set used in training them. We trained the M\_6150 model using 6150 samples. This consisted of 2050 samples from each of the classes. The CRC class was augmented to 3000, and three thousand samples were randomly from PTW class and the NOISE class to give a total of 9000 samples which were used to train the M\_9000 model. Next, CRC was augmented to 4000 samples and the same sample sizes selected from the PTW and NOISE classes. This gave a total of 12000 samples. A similar method was applied to create a training set of 16000 samples that were used to train the M\_16000 model. Finally, we used 21000 samples, 7000 samples from each class to train the M\_21000 model. Each of the 5 baseline models was executed separately on their corresponding training subsets. Each model took an average of 10 minutes to

train. The results obtained for the various experiments are shown in figure 5.3, figure 5.4 and table 5.2.

Similar to the pre-trained models, the performances of the baseline models increase as the size of the training subset increases. Figure 5.3 shows the boxplot of the average accuracy; M\_21000 model produced the highest accuracy while M\_9000 produced the least accuracy. Again, the boxplot in figure 5.4 indicates that M\_21000 model had the highest F1 score while M\_9000 model had the lowest F1 score. It shows that as we increase the augmentation process the results improve, and the variation decreases. It means that future work should augment their data to improve performance. It is worth mentioning that the boxplots were created in R.

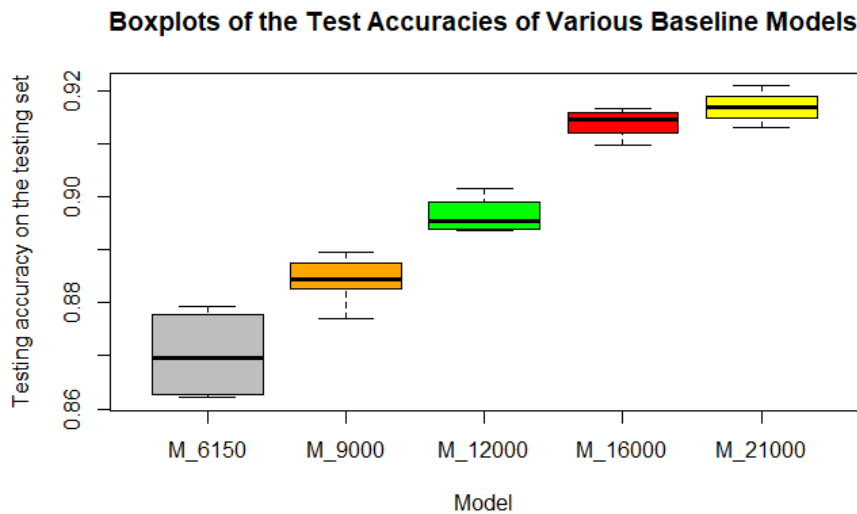


Figure 5.3: Comparison of accuracy of various baseline models

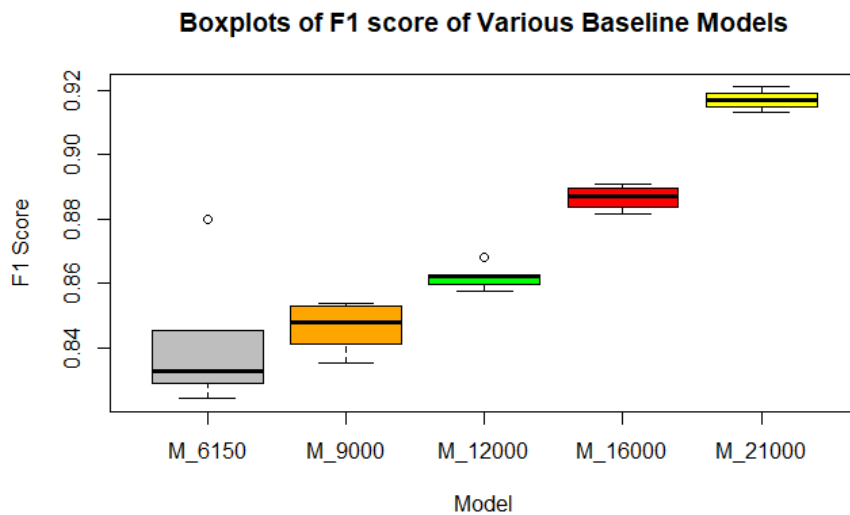


Figure 5.4: Comparison of F1 score of various baseline models

Table 5.2: Average performance measures of various baseline models.

Average performance measures of various baseline Models					
Measure	M_6150	M_9000	M_12000	M_16000	M_21000
Recall	0.8736	0.8763	0.8946	0.8903	0.9045
Precision	0.8207	0.8260	0.8393	0.8845	0.8734
F1 Score	0.8406	0.8464	0.8621	0.8865	0.8872
AUC	0.9530	0.9600	0.9650	0.9650	0.9690

### 5.3 Fine-tuning

The pre-trained models were fine-tuned using the different subsets of the training set: we used 9000, 12000, 16000, and 21000 spectrograms. Table 5.3 depicts the result of the model fine-tuned with 9000 samples, 3000 sam-



ples from each class.

Table 5.3: Average performances of various pre-trained models fine-tuned with 9000 spectrograms.

Pre-trained models fine-tuned with 9000 spectrograms			
Measure	PT-9000	PT-12000	PT-15000
Accuracy	0.8712	0.8790	0.8845
Precision	0.8096	0.8190	0.8308
Recall	0.8830	0.8859	0.8820
F1 score	0.8355	0.8446	0.8516
AUC	0.96151	0.9602	0.9590

Then, the pretrained models were each fine-tuned with 12000 samples, 4000 samples from each class. The results obtained are shown in table 5.4.

Table 5.4: Average performances of various pre-trained models, fine-tuned with 12000 spectrograms.

Pre-trained models fine-tuned with 12000 spectrograms			
Measure	PT-9000	PT-12000	PT-15000
Accuracy	0.8922	0.8889	0.8943
Precision	0.8406	0.8347	0.8387
Recall	0.8888	0.8347	0.8897
F1 score	0.8605	0.8593	0.8596
AUC	0.9641	0.9681	0.9614

Next, we fine-tuned the pretrained models with 16000 samples, 5000 CRC

spectrograms, 5315 PTW spectrograms, and 5685 NOISE spectrograms. Table 5.5 shows the average result obtained.

Table 5.5: Average performances of various pre-trained models, fine-tuned with 16k spectrograms.

Pre-trained models fine-tuned with 16k spectrograms			
Measure	PT-9000	PT-12000	PT-15000
Accuracy	0.9057	0.9106	0.9090
Precision	0.8665	0.8748	0.8707
Recall	0.8876	0.8926	0.8882
F1 score	0.8755	0.8829	0.8782
AUC	0.9600	0.9700	0.9600

Finally, the results in table 5.6 were obtained by fine-tuning the different pre-trained model using 21000 spectrograms. CRC and PTW classes were each augmented to 7000 spectrograms and 7000 spectrograms obtained from the NOISE class.

Table 5.6: Average performances of various pre-trained models, fine-tuned with 21k spectrograms.

Pre-trained Models fine-tuned with 21k spectrograms			
Measure	PT-9000	PT-12000	PT-15000
Accuracy	0.9164	0.9158	0.9173
Precision	0.8870	0.8823	0.8849
Recall	0.8897	0.9000	0.8933
F1 score	0.8878	0.8897	0.8930
AUC	0.9616	0.97	0.963

## 5.4 Comparison of baseline models and fine-tuned models

In this section, we compare the baseline models with the pre-trained and fine-tuned models. Both baseline models and the fine-tuned models were evaluated using the testing set from collected data. It is, therefore, reasonable to compare their performances. We use the F1 score and test accuracy to compare them. Figure 5.6 shows the comparison. Baseline model and fine-tuned model trained using the same number of spectrograms are placed next to each other, fine-tuned model first. For instance, the baseline model trained using 9000 (M\_9000) spectrograms is placed next to the model fine-tuned using 9000 spectrograms (FT-9000) for both metrics used.

Considering the F1 score, the models fine-tuned using 9000, 16000, and 21000 spectrograms outperformed the baseline models trained on the same number of spectrograms. However, the baseline model trained using 12000 (F1 score of 0.8960) outperformed the models fine-tuned using the same (F1 score of 0.8943). It is clear from figure 5.6 that the baseline models produced better testing accuracy than the fine-tuned models. In 5.6c, the baseline model trained with 16000 spectrograms has a higher accuracy than the fine-tuned model trained with 16000. When a pre-trained model was fine-tuned with 21000 spectrograms, and compared with a baseline model trained with the same quantity of spectrograms, both accuracy and the F1 score of the baseline model was higher than that of the fine-tuned model as seen in figure 5.6d. Further comparison of the baseline and the fine-tuned is shown in figure 5.5, and it is observed that the baseline models outperformed the fine-tuned models.

Our result shows that augmentation by time-shifting increases the performance of the models; as we increase the number of samples by data augmentation, the performance of the pre-trained models, baseline models, and the fine-tuned model increase. Thus, time-shifting data augmentation increases the performance of bioacoustic classifiers. This is in line with result obtained by Dufourq et al. (2021).

The baseline models performed better than the fine-tuned models. A potential explanation for this result is that the data from Xeno-canto is somehow slightly different from the data that we collected in that our data had a constant sampling rate while the data from Xeno-canto had varying sampling rates based on each person’s microphone. That is, there was no correlation between the performance of the pretrained models and the site of training of fine tuning the pre-trained models. Another reason could be that the training set used in pretraining was not big enough.

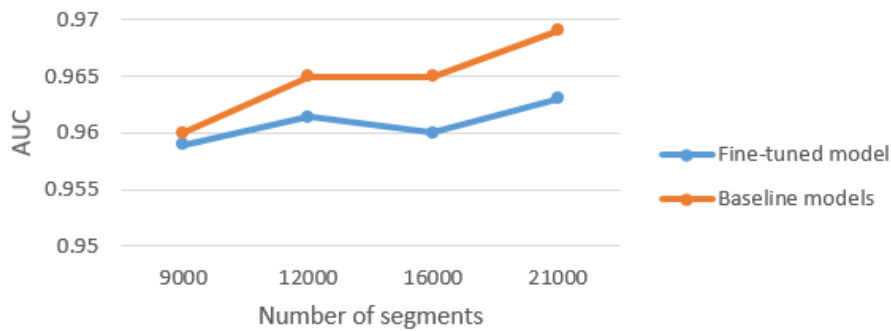
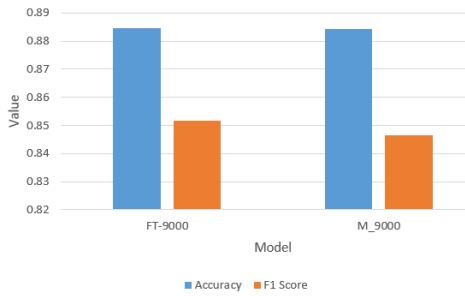
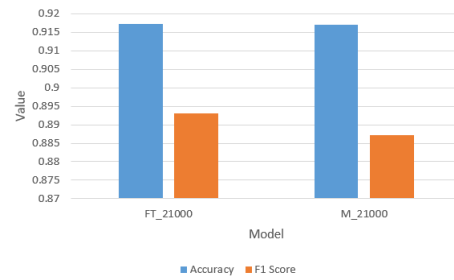


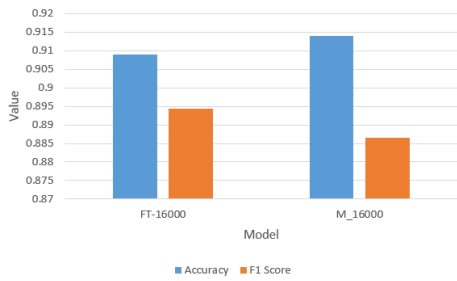
Figure 5.5: Comparing of baseline and fine-tuned models using AUC



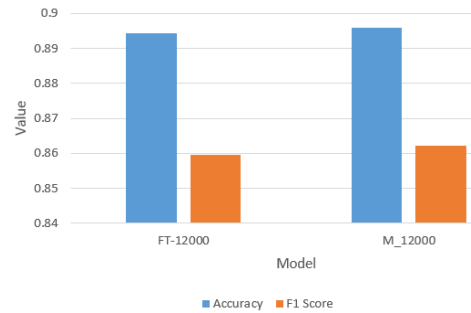
(a) Comparison of baseline model and fine-tuned model trained using 9000 spectrograms.



(b) Comparison of baseline model and fine-tuned model trained using 12000 spectrograms.



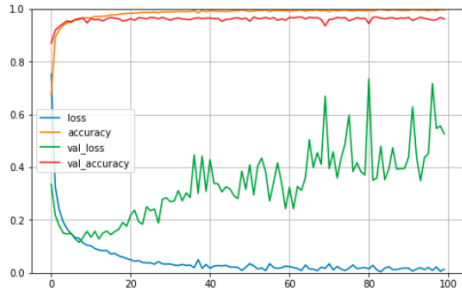
(c) Comparison of baseline model and fine-tuned model trained using 16000 spectrograms.



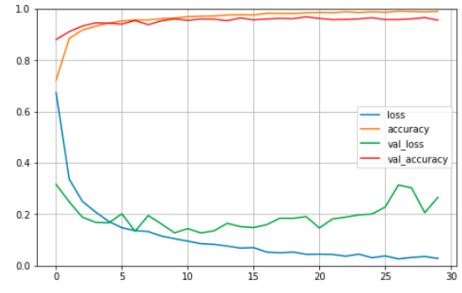
(d) Comparison of baseline model and fine-tuned model trained using 21000 spectrograms.

Figure 5.6: Comparison of baseline models with pre-trained and fine-tuned models using different samples sizes. FT-9000 model refers to the model fine-tuned using 9000 samples, 3000 samples from each of the three classes and M\_9000 represents a baseline model trained using 9000 samples, 3000 samples from each of the classes. FT-12000 model represents a pre-trained model fine-tuned using 12000, M\_12000 represents the baseline model trained using 12000. The same applies to FT-16000, M\_16000, FT-21000, and M\_21000 models.

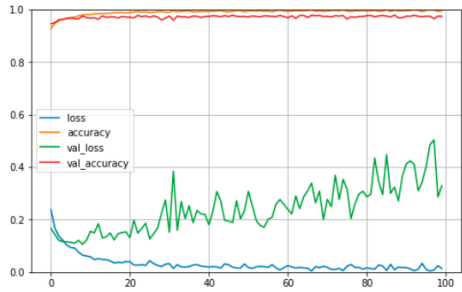
We monitored our models using learning curves. A learning curve is a plot of training loss and or accuracy and validation loss and or accuracy against the number of training epochs. Figure 5.7 shows the learning curves for baseline models compared with the learning curve of the pre-trained and fine-tuned models. Figure 5.7a shows a baseline model that was trained on the entire training set without augmentation and early stopping. The model was greatly overfitted. In figure 5.7b, we employed early stopping to reduce/prevent overfitting by preventing model from learning noise. Thus, early stopping prevented noise from affecting the model. Our results show that early stopping reduces overfitting. Overfitting was further reduced using a dropout rate of 0.5. Figure 5.7c and figure 5.7d show the learning curves of one of the fine-tuned models. Figure 5.7c is the learning curve of a pre-trained model which was fine-tuned on augmented data without early stopping. Still, the model overfitted, but not much as the baseline model. Similarly, figure 5.7d shows the learning curve of the same model as c with early stopping. It was observed that data augmentation, early stopping, and dropout prevent overfitting. Therefore, increasing data size through data augmentation is an effective method to prevent overfitting in bioacoustics. That is, data augmentation is a good technique to solve the data limitation problem which will consequently prevent overfitting. Early stopping and dropout also prevent overfitting.



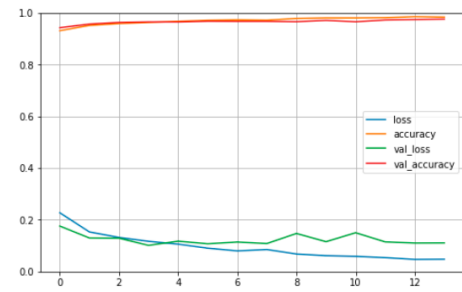
(a) Learning curve for baseline model with augmentation without early stopping.



(b) Learning curve for baseline model with augmentation with early stopping.



(c) Learning curve for fine-tuned model trained using 21000 spectrograms without early stopping



(d) learning curve for fine-tuned model trained using 21000 spectrograms with early stopping

Figure 5.7: Learning curves for the baseline model without augmentation with and without early stopping.

# Chapter 6

## Conclusion

This research was set out to build CNNs for bioacoustic classifiers using bird vocalizations. Currently, there are some limitations regarding the use of machine learning algorithms for bioacoustic monitoring. At the moment, researchers manually annotate all data; this is extremely time-consuming and unsustainable. This is not feasible if one has a large number of audio records, each lasting for long hours. Also, there is limited vocalization data for certain bird species. Some birds do not call often, and it is challenging to obtain recordings of birds whose habitats are inaccessible. Furthermore, it is difficult to obtain enough data about endangered species even though some data of their vocalizations might exist. These recordings are protected and not made publicly available. This is sometimes done to protect the species as poachers might want to go and capture/kill the species. Training CNNs with such a limited amount of data may result in overfitting. However, trained CNN classifiers that can automatically classify bird species based solely based on their vocalizations will facilitate the monitor-



ing of birds by ecologists and conservationists by reducing time and cost. One other reason why we carried out this study was because of its novelty; although transfer learning has been used in bioacoustics, no pre-trained model has been built using bird vocalization from Xeno-canto and eBird and fine-tuned using primary data.

Firstly, our secondary data was downloaded from Xeno-canto and eBird. Secondly, primary data was collected from Intaka Island Nature Reserve, and then both data were preprocessed. Preprocessing involved the annotation of audio files and the conversion of the annotated audio into spectrograms. Next, CNN networks were trained using spectrograms. Secondary data was used to pretrain three models while the primary data was used for building the baseline models and fine-tuning the pre-trained models. We successfully pre-trained three models (PT-9000, PT-12000, PT-16000) using vocalization of two bird species- pin-tailed whydah (*Vidua macroura*) and Cape robin-chat (*Cossypha caffra*) using audio recording from Xeno-canto and eBird repositories. In addition, we built five baseline models using bird vocalization obtained from Intaka Island, Cape Town which performed well in the discrimination of spectrograms into PTW, CRC, and NOISE. Finally, the pre-trained models were fine-tuned using different subsets of training data obtained from the collected data.

The time-shifting augmentation techniques greatly improved both baseline and pre-trained models' performance. Our fine-tuned model obtained almost equal performance; however, the baseline models slightly outperformed the fine-tuned models. The best baseline model (M\_21000) had a test accuracy of 91.70% while the best fine-tuned model achieved 91.73%. The AUC for the best baseline was 96.9% against 96.3% for the best fine-

tuned model (FT-21000). These results could be because the data used for pre-training was not big enough. We used Xeno-canto and eBird which did not have a large number of audio recordings on the species of interest at the time when this study was done. In addition, we observed that the overfitting problem which usually results from training CNN with small amounts of data can be greatly reduced or prevented using time-shifting data augmentation technique, early stopping, and or dropout techniques. Finally, we published a data set on zenodo for public use. The link to the data set found [here](#).

## 6.1 Future work

We used only the CNN algorithm in this dissertation, and future work could be to use a combination of CNN and recurrent neural networks. It is hoped that this combination will achieve better outcomes. Given that the total number of spectrograms used in pre-training was not large enough, we could obtain better performance by considering other bird vocalization repositories where we can obtain a large data size. Another way to improve our pre-trained models is to use more bird species in our subsequent work. Furthermore, we will assess the impact of the quality of recordings on our classification models by splitting the data obtained from Xeno-canto into two subsets: high quality and low-quality categories. Finally, in the future, we intend to deploy our best model.

## 6.2 Recommendation

The performances of our deep learning models were excellent, and they indicate that deep learning is invaluable in the monitoring and conservation of ecology. We showed that the time-shifting augmentation technique is a good technique to augment data in bioacoustic. Hence, they solve the problem of data limitation, and this will also help to prevent overfitting. Our models performed well in distinguishing the various species. It means they can be readily employed to complement the manual automation process, making bioacoustic monitoring less expensive, less time-consuming, and sustainable. Therefore, our model can be used in biodiversity in the monitoring and conservation of the ecology.

# References

- Acevedo, M. A., Corrada-Bravo, C. J., Corrada-Bravo, H., Villanueva-Rivera, L. J., & Aide, T. M. (2009). Automated classification of bird and amphibian calls using machine learning: A comparison of methods. *Ecological Informatics*, *4*(4), 206–214.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6.
- Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology*, *12*(7), 878.
- Ayodele, T. O. (2010). Types of machine learning algorithms. *New advances in machine learning*, *3*, 19–48.
- Bianco, M. J., Gerstoft, P., Traer, J., Ozanich, E., Roch, M. A., Gannot, S., & Deledalle, C.-A. (2019). Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, *146*(5), 3590–3628.
- Bottou, L. (2012). Stochastic gradient descent tricks. *Neural networks: Tricks of the trade* (pp. 421–436). Springer.
- Brauer, P. D. J. (2018). *Introduction to deep learning*.

- Cakir, E., Adavanne, S., Parascandolo, G., Drossos, K., & Virtanen, T. (2017). Convolutional recurrent neural networks for bird audio detection. *2017 25th European Signal Processing Conference (EU-SIPCO)*, 1744–1748.
- Campbell, M., Hoane Jr, A. J., & Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, *134*(1-2), 57–83.
- Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology and Evolution*, *10*(10), 1632–1644.
- D. Rosa, I. M., Marques, A. T., Palminha, G., Costa, H., Mascarenhas, M., Fonseca, C., & Bernardino, J. (2016). Classification success of six machine learning algorithms in radar ornithology. *Ibis*, *158*(1), 28–42.
- D’Angelo, G., Tipaldi, M., Glielmo, L., & Rampone, S. (2017). Spacecraft autonomy modeled via markov decision process and associative rule-based machine learning. *2017 IEEE international workshop on metrology for aerospace (MetroAeroSpace)*, 324–329.
- Darras, K., Furnas, B., Fitriawan, I., Mulyani, Y., & Tscharrntke, T. (2018). Estimating bird detection distances in sound recordings for standardizing detection ranges and distance sampling. *Methods in Ecology and Evolution*, *9*(9), 1928–1938.
- Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al. (2012). Large scale distributed deep networks. *Advances in neural information processing systems*, *25*, 1223–1231.

- Debnath, S., Roy, P. P., Ali, A. A., & Amin, M. A. (2016). Identification of bird species from their singing. *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, 182–186.
- Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., et al. (2013). Recent advances in deep learning for speech research at microsoft. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8604–8608.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- Dufourq, E., Durbach, I., Hansford, J. P., Hoepfner, A., Ma, H., Bryant, J. V., Stender, C. S., Li, W., Liu, Z., Chen, Q., et al. (2021). Automated detection of hainan gibbon calls for passive acoustic monitoring. *Remote Sensing in Ecology and Conservation*.
- El Naqa, I., & Murphy, M. J. (2015). What is machine learning? *Machine learning in radiation oncology* (pp. 3–11). Springer.
- Fagerlund, S. (2007). Bird species recognition using support vector machines. *EURASIP Journal on Advances in Signal Processing*, 2007, 1–8.
- Ferrucci, D. A. (2012). Introduction to “this is watson”. *IBM Journal of Research and Development*, 56(3.4), 1–1.
- Gee, J. P. (2009). Deep learning properties of good digital games: How far can they go? *Serious games* (pp. 89–104). Routledge.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

- Hauster, J. E. (2015). *How can bioacoustics help conserve biodiversity?*  
Retrieved April 5, 2021, from <https://blog.nature.org/science/explainer/how-can-bioacoustics-help- conserve-biodiversity/>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 3–12.
- Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.
- Hill, A. P., Prince, P., Snaddon, J. L., Doncaster, C. P., & Rogers, A. (2019). Audiomoth: A low-cost acoustic device for monitoring biodiversity and the environment. *HardwareX*, 6, e00073.
- Hinton, G. (2012). *Lecture 6d: A separate, adaptive learning rate for each connection. slides of lecture neural networks for machine learning* (tech. rep.). Technical report, Slides of Lecture Neural Networks for Machine Learning.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Hossin, M., & Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

- Incze, A., Jancsó, H.-B., Szilágyi, Z., Farkas, A., & Sulyok, C. (2018). Bird sound recognition using a convolutional neural network. *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, 000295–000300.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.
- Kampichler, C., Wieland, R., Calmé, S., Weissenberger, H., & Arriaga-Weiss, S. (2010). Classification in conservation biology: A comparison of five machine-learning methods. *Ecological Informatics*, 5(6), 441–450.
- Ketkar, N. (2017). *Deep learning with python, a hands-on introduction*, apress edition, 160p.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Konečný, J., Liu, J., Richtárik, P., & Takáč, M. (2015). Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2), 242–255.
- Lake Tahoe, U., Nevada. (2013). *Neural information processing scaled for bioacoustics: Nips4b*. Retrieved July 4, 2021, from <http://sabiody.univ-tln.fr/nips4b/challenge1.html>
- Landau, H. (1967). Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10), 1701–1706.
- Lasseck, M. (2013). Bird song classification in field recordings: Winning solution for nips4b 2013 competition. *Proc. of int. symp. Neural Information Scaled for Bioacoustics, sabiod. org/nips4b, joint to NIPS, Nevada*, 176–181.



- Lasseck, M. (2015). Improved automatic bird identification through decision tree based feature selection and bagging. *CLEF (Working Notes)*.
- Leng, Y. R., Dennis, J. W., & Dat, T. H. (2014). Bird classification using ensemble classifiers. *CLEF (Working Notes)*, 654–661.
- Limonova, E., Alfonso, D., Nikolaev, D., & Arlazarov, V. V. (2021). Resnet-like architecture with low hardware requirements. *2020 25th International Conference on Pattern Recognition (ICPR)*, 6204–6211.
- Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231.
- Mitchell, T. M. et al. (1997). Machine learning.
- Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., & Melgani, F. (2019). Computer vision-based phenotyping for improvement of plant productivity: A machine learning perspective. *GigaScience*, *8*(1), giy153.
- NarasingaRao, M., Venkatesh Prasad, V., Sai Teja, P., Zindavali, M., & Phanindra Reddy, O. (2018). A survey on prevention of overfitting in convolution neural networks using machine learning techniques. *Int. J. Eng. Technol*, *7*(2.32), 177–180.
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.
- Priyadarshani, N., Castro, I., & Marsland, S. (2018). The impact of environmental factors in birdsong acquisition using automated recorders. *Ecology and evolution*, *8*(10), 5016–5033.
- Ramashini, M., Abas, P. E., Grafe, U., & De Silva, L. C. (2019). Bird sounds classification using linear discriminant analysis. *2019 4th Interna-*

- tional Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, 1–6.
- Roodschild, M., Sardiñas, J. G., & Will, A. (2020). A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4), 351–360.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210–229.
- Sankupellay, M., & Konovalov, D. (2018). Bird call recognition using deep convolutional neural network, resnet-50. *Proceedings of ACOUSTICS*, 7(9).
- Sarle, W. S. (1996). Stopped training and other remedies for overfitting. *Computing science and statistics*, 352–360.
- Sharma, S. (2017). Activation functions in neural networks. *towards data science*, 6.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. *arXiv preprint arXiv:2007.14166*.
- Sprengel, E., Jaggi, M., Kilcher, Y., & Hofmann, T. (2016). *Audio based bird species identification using deep learning techniques* (tech. rep.).
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from

- overfitting. *The journal of machine learning research*, 15(1), 1929–1958.
- Steiner, W. W. (1981). Species-specific differences in pure tonal whistle vocalizations of five western north atlantic dolphin species. *Behavioral Ecology and Sociobiology*, 9(4), 241–246.
- Stowell, D., & Plumbley, M. D. (2013). *Freefield1010 - an open dataset for research on audio field recording archives*. Retrieved July 5, 2021, from <https://arxiv.org/abs/1309.5275>
- Stowell, D., Wood, M. D., Pamuła, H., Stylianou, Y., & Glotin, H. (2019). Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge. *Methods in Ecology and Evolution*, 10(3), 368–380.
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. *International conference on artificial neural networks*, 270–279.
- Thomas, M., Martin, B., Kowarski, K., Gaudet, B., & Matwin, S. (2019). Marine mammal species classification using convolutional neural networks and a novel acoustic representation. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 290–305.
- Torrey, L., & Shavlik, J. (2010). Transfer learning. *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques* (pp. 242–264). IGI global.
- Tóth, B. P., & Czeba, B. (2016). Convolutional neural networks for large-scale bird song classification in noisy environment. *CLEF (Working Notes)*, 560–568.

- Trifa, V. M., Kirschel, A. N., Taylor, C. E., & Vallejo, E. E. (2008). Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *The Journal of the Acoustical Society of America*, *123*(4), 2424–2431.
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.
- Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, *10*(5), 1897.
- Xie, J.-j., Ding, C.-q., Li, W.-b., & Cai, C.-h. (2018). Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks. *arXiv preprint arXiv:1803.01107*.
- Xu, K., Cai, H., Liu, X., Gao, Z., & Zhang, B. (2017). North atlantic right whale call detection with very deep convolutional neural networks. *The Journal of the Acoustical Society of America*, *141*(5), 3944–3945.
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligence magazine*, *13*(3), 55–75.
- Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E.-D., Jin, W., & Schuller, B. (2018). Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *9*(5), 1–28.



# Appendix A

## Plagiarism Report

PRE-TRAINING NEURAL NETWORKS ON XENO-CANTO AND  
EBIRD FOR BIOACOUSTIC CLASSIFICATION MODEL

ORIGINALITY REPORT

18%

SIMILARITY INDEX

12%

INTERNET SOURCES

12%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	dr.ur.ac.rw Internet Source	1%
2	Emmanuel Dufourq, Carly Batist, Ruben Foquet, Ian Durbach. "Passive acoustic monitoring of animal populations with transfer learning", Ecological Informatics, 2022 Publication	1%
3	asa.scitation.org Internet Source	1%
4	Batuhan Yilmaz, Melih Sen, Engin Masazade, Vedat Beskardes. "chapter 4 Behavior Classification of Egyptian Fruit Bat (Rousettus aegyptiacus) From Calls With Deep Learning", IGI Global, 2022 Publication	<1%
5	Mohammad Alkhaleefah, Shang-Chih Ma, Yang-Lang Chang, Bormin Huang, Praveen Kumar Chittam, Vishnu Priya Achhannagari. "Double-Shot Transfer Learning for Breast 72	<1%