



**AFRICAN CENTRE OF EXCELLENCE
IN DATA SCIENCE**



COLLEGE OF BUSINESS & ECONOMICS

Acoustic Data Augmentation for Small Passive Acoustic Monitoring Datasets

By

NSHIMIYIMANA Aime

Registration Number: 220003483

**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: **Dr. Emmanuel Dufourq**

September 2022

Declaration

I declare that this dissertation entitled “Acoustic Data Augmentation for Small Passive Acoustic Monitoring Datasets” 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.

Candidate:



28 June 2022


NSHIMIYIMANA Aime

Ref NO:220003483

Date

Approval

This dissertation entitled “Acoustic Data Augmentation for Small Passive Acoustic Monitoring Datasets” written and submitted by NSHIMIYIMANA Aime 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 10% which is less than 20% accepted by the African Centre of Excellence in Data Science (ACE-DS).

Supervisor: 

Dr. Emmanuel Dufourq

28 June 2022

Date

Head of training: _____

Date

Dedication

I dedicate this thesis to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge, and understanding. He has been the source of my strength throughout this program and on his wings only have I soared.

Acknowledgements

First and foremost I am extremely grateful to my supervisor, Dr. Emmanuel Dufourq for his invaluable advice, continuous support, and patience during my postgraduate research. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life activities. I would like to recognize his invaluable assistance of python scripts for audio pre-processing¹ and reading annotations from audio data².

My gratitude extends to the higher educational council (HEC Rwanda) for the funding opportunity to undertake my studies at the center of excellence in data science, the University of Rwanda. I would like to thank Intaka Island Nature Reserve, Cape Town, South Africa for granting access and allowing us to record audio.

I would like to thank my colleague and classmate, Mr. Mikwa Boris for helping with the typesetting and grammatical errors of the thesis writing. We thank EdgeAcoustics NPO for providing support in collecting the audio data and for providing the necessary training in bioacoustics research.

I wish to express my sincere appreciation to my supervisor Dr. Emmanuel and my friend Mr. Boris for annotated and verified the audio data together. This was a huge effort that took days of work with all three of us. I would like to pay my special regards to Mark Heerden for funding AudioMoths and other audio equipment which we used in this research.

¹Python scripts for audio pre-processing <https://github.com/emmanueldufourq/MLEcology/blob/main/ProcessAudioFiles.ipynb>

²Python scripts for reading annotations from audio <https://github.com/emmanueldufourq/MLEcology/blob/main/AnnotationReader.py>

Abstract

Training complex deep neural networks can result in overfitting when the networks are trained from random weight initialization on small datasets. Data augmentation helps to reduce the negative effects of overfitting. Data augmentation is the process by which the amount of data for a given problem is increased in quantity via some augmentation technique. The findings in computer vision and audio recognition research reveals that the performance of machine learning classifiers is significantly improved when the data is augmented.

In the context of ecology, researchers conduct field surveys whereby microphones are placed in some location and audio data is recorded over a period of time. There is however no guarantee that the particular species of interest in the field survey will vocalize frequently near the microphone. Thus, the amount of data captured for the species of interest might be limited. Training robust classifier models on such limited data will most likely lead to overfitting.

The purpose of this research is to investigate several audio augmentation techniques as a means to increase the amount of audio examples for certain species of interest with the goal of creating robust audio vocalization classifier models. We investigate noise injection and time and frequency masking data augmentation techniques. These techniques are applied to two birds of interest, namely the pin-tailed whydah (*Vidua macroura*) and the Cape robin-chat (*Cossypha caffra*). While these two species are not endangered, they allow us to compare the various augmentation techniques. The audio recordings were obtained from the Intaka Island Nature Reserve, South Africa.

To evaluate the performance of the augmentation techniques we conducted a comparison between experiments run with and without augmentation. We chose to use convolutional neural networks as our classifier given that they are the state-of-the-art in audio recognition tasks. Furthermore, convolutional neural networks have revealed good performance in the field of bioacoustics.

We manually annotated 768 audio files (20 minutes each) totaling over 256 hours of

audio. We labeled the start and stop time for three events (segments of audio which contained calls of the pin-tailed whydah, Cape robin-chat, or no calls of either two species) and thus created a three-class audio classification problem. The experiment of the study only considered 162 audio files (54 hours) due to computational limitations.

We intentionally randomly sampled 500 examples which contained the two species calls from the training data and then augmented this to 2,000 examples. The results revealed that in doing so the network obtained 90.2% testing accuracy using time masking as the augmentation technique. This finding is important as it reveals that convolutional neural networks can be used even when the amount of training data is as small as 500 examples, and thus assist in machine learning research for where audio data might be limited for the species being surveyed.

Keywords: Data augmentation; Bioacoustics; Deep learning

Contents

Declaration	I
Approval	II
Dedication	III
Acknowledgements	IV
Abstract	V
Contents	VII
List of Figures	IX
List of Tables	XI
List of abbreviations	XII
Chapter 1 Introduction	1
1.1 Rationale	1
1.2 Background of the research	3
1.2.1 Problem statement	5
1.2.2 Objectives	6
1.3 Organization of the research	7
Chapter 2 Literature review	8
2.1 Introduction	8
2.2 Related work	9

Chapter 3 Methodology	16
3.1 Data Collection	16
3.1.1 Pre-processing	19
3.1.2 Data after the pre-processing phase	25
3.2 Machine Learning	25
3.3 Deep learning	26
3.3.1 Artificial Neural Networks	26
3.3.2 Activation functions	28
3.3.3 Convolutional Neural Networks	33
3.3.4 Convolutional layers	34
3.3.5 Max pooling	35
3.3.6 Fully connected layer	36
3.3.7 Evaluating model performance	36
3.4 Data augmentation	39
3.4.1 Baseline method	41
3.4.2 Noise addition	41
3.4.3 Masking	44
Chapter 4 Results and discussion	46
4.1 Introduction	46
4.2 Model training and hyper-parameter tuning	46
4.3 Results of the findings and their comparison	49
Chapter 5 Conclusions and recommendations	54
5.1 Conclusions	54
5.2 Recommendations	55
References	56

List of Figures

0.1	Intaka Island Nature Reserve, Cape Town, South Africa.	16
0.2	Adult male pin-tailed whydah, extracted from eBird 1231212312	18
0.3	Cape robin-chat, extracted from eBird 1231212312	18
0.4	An attachment of AudioMoth recorder to the tree.	19
0.5	An illustration of a spectrogram image.	21
0.6	An illustration of a waveform image.	22
0.7	Sonic visualizer annotation images.	22
0.8	Another call of Cape robin-chat call image	23
0.9	Example of pin-tailed whydah call images.	23
0.10	Typical input image of pin-tailed whydah call to CNNs model	23
0.11	Typical input image of Cape robin-chat call to CNNs model	24
0.12	ANN with one input layer, two hidden layers and one output layer.	27
0.13	Kernel example with stride of 1 (Input, kernel and output).	34
0.14	Example of max pooling layer	36
0.15	original image of pin-tailed whydah before augmentation	42
0.16	Random noise addition applied to pin-tailed whydah	42
0.17	Gaussian noise addition applied to pin-tailed whydah	43
0.18	Time masking applied to pin-tailed whydah	45
0.19	Frequency masking applied to pin-tailed whydah	45
0.1	Frequency masking evaluation plot for 100 sample.	52
0.2	Time masking evaluation plot for 100 sample.	52
0.3	Baseline evaluation plot for 100 sample.	52

0.4	Gaussian noise addition	
	evaluation plot for 100 sample.	52
0.5	Random noise evaluation	
	plot for 500 sample.	53
0.6	Gaussian noise evaluation	
	plot for 500 sample.	53

List of Tables

- 0.1 Confusion matrix 38
- 0.1 Training CNN model parameters. 48
- 0.2 Comparison between experiment techniques versus sample size. 50

List of abbreviations

1D:1 dimension

2D:2 dimension

ACE:Automatic content extraction

ANN:Artificial neural network

ARS:Augmented random search

CIFAR-10:Canadian institute for advanced research, 10 classes

CELM:Convolutional extreme learning machine

CNN:Convolutional neural network

CSVM:Convolutional support vector machine

DADA:Deep adversarial data augmentation

DAG:Data augmentation using generative adversarial networks

DDSM:Digital database for screening mammography

DGGAN:Deep convolutional generative adversarial network

DL:Deep learning

ELU:Exponential linear unit

FER2013:Facial expression database 2013

FN:False negative

FP:False positive

GAN:Generative adversarial networks

GMM:Gaussian mixture models

HEC:Higher education council

HMM:Hidden markov models

KNN:K-nearest neighbor

LSTM:Long short-term memory

MAE:Mean absolute error

MATLAB:Matrix laboratory

MBE:Mean bias error

MF:Mean fold model
MLP:Multi-layer perceptron
MNIST:Modified national institute of standards and technology database
MRR:Reciprocate rank in percentage
MSE:Mean squared error
NPV:Negative predictive value
PReLU:Parametric ReLU
ReLU:Rectified linear unit
RF:Random forest
RNN:Recurrent neural networks
RReLU:Reflected rectified linear unit
SD:Secure digital
SVM:Support vector machine
Tanh:Tangent hyperbolic
TN:True negative
TP:True positive
WGAN:Wasserstein GAN

Chapter 1 Introduction

1.1 Rationale

Research in acoustics deals with the science of sounds and, it is becoming the most trend of science and engineering insights in bioacoustics in recent years (Bianco et al., 2019; Teixeira, Maron, & van Rensburg, 2019). The term bioacoustics is defined as the study of living animals, their sounds production, transmission, and reception.

Conservation and monitoring of species are the main tasks of ecologists in the field of bioacoustics. This involves the animal vocalization, detection, identification, tracking of species, and measuring the population density of species in the region (Mcloughlin, Stewart, & McElligott, 2019). Conservation relates to the human activities of shipping species or damaging effect versus living species in the environment (Mcloughlin et al., 2019). Vocalization and species call are fundamental elements for monitoring and conservation of animals.

Vocalization measures emotional, physiological, and individual animal behavior (Mcloughlin et al., 2019). Behaviors of domestic animals express their health condition and welfare. The vocal sound of the goat is an example to explain negative and positive experience behavior (Mcloughlin et al., 2019). Alarm calls from eagles express the presence of predator animals (Seyfarth & Cheney, 2003). Vocalization of animals such as the growling of dogs and the roaring of lions indicate the rage of these animals, therefore, they pretend to strike against their enemies or making them have a lot of fear (Seyfarth & Cheney, 2003). Vocalization also is used to determine the population size for each individual species, call detection and their classification.

Species detection deals with the identification of the presence of animal sound in the audio recordings. Detection and classification of species find specific patterns in the call

spectrogram (Nanni, Maguolo, & Paci, 2020; Tóth & Czeba, 2016; Çakir, Adavanne, Parascandolo, Drossos, & Virtanen, 2017). A spectrogram is the visual representation of the signal (e.g. sound) frequencies as it changes with time. The studies mentioned in this paragraph have all provided the success of the audio classification using the spectrograms representations to the audio data. Sound analysis is achieved by using the computer vision (image analysis and classification) approach in machine learning. In the call detection segments of call/song are isolated for specific species in the entire recordings.

Generally, the annotation is used to label training examples which are then used to train the classifier to find the calls. In our case, the location of the call is determined by labeling and putting a surrounding box on the call during the period of recordings. Source localization is another task used in vocalization and it is tracking, assessing spectrograms of calls for particular species in the recordings.

The population of species takes into accounts the density, abundance, size, or number of species in habitat space or surrounding environment. Call count is the basic approach for species population estimation. This technique is complex and requires a lot of effort and expertise in the field of bioacoustics. Finding the population of birds species is one example of complex activity. When people walk around and make noises, this can cause a bird to stop vocalising and fly away. Therefore results about their population may be biased with a human being present in the space. It is good practice to be out of sight to get more accurate data from bird species. The use of audio recorders helps to solve the problem of the presence of human beings since these recorders are non-invasive and passive acoustic monitoring techniques (Browning, Gibb, Glover-Kapfer, & Jones, 2017). Hence findings from collected data of species will have unbiased results (Priyadarshani, Marsland, & Castro, 2018).

Audio recorders provide automatic data acquisition of not only the birds but also other species in an effective and convenient way. Calls and vocal sounds of the birds are taken without extra human being efforts. People simply mount and leave recorders in the trees for the predefined specific time periods for capturing data. Recorders are also set to the maximum frequency of recording. This frequency allows the caption for species calling at the most frequency specified at the recorder during the setup. The

time and frequency of the call to be captured will depend on the recorder's settings. Recorders are mainly used for collecting data without the intervention of humans at the sight, they capture sounds of data for an extensive time period. The recorders are also used to capture data from hard areas to access/complex environment space. Recorders can be programmed to turn on or off at given times. Thus, this allows researchers to capture audio recordings at early hours in the day, or even at night. This enables the collection of diurnal and nocturnal sounds of species (Priyadarshani et al., 2018).

The recorders are installed in the trees for capturing birds calls and other sounds within the range of frequency specified by the recorder. These recorders have the capability to record audio species for long periods of time (Bianco et al., 2019). If there could be endangered species, their calls are also captured during the same time. The process of audio analysis and classification starts with data collection. It proceeds with data preparation and data pre-processing.

The process of data collection is not enough for analysis. It is followed by data pre-processing, call detection, and segmentation. Convolutional neural networks (CNN) can learn filters which can result in good audio classification performance. These tasks are currently being performed manually. This is complex and time-consuming work. The need is to automate current ecologist tasks and the process of analysis for call detection and classification based on audio recordings. These automatic processes and tasks will speed up operations and computations. In this thesis we explore the automation of vocalization classification on two species (pin-tailed whydah and Cape robin-chat), however, one could also apply the same approach to other species. We know that birds play an important role for an indication of any major change to the environment (Koskimies, 1989) then it is easy to detect changes in the environment by observing their composition, quantity, and biodiversity to reflect habitat suitability. This is the reason why the study is based on the bird vocalisations.

1.2 Background of the research

Current researchers in the field of bioacoustics report deep learning as the most successful for identification of bird species (Sankupellay & Konovalov, 2018; jian Xie, qing Ding, Li, & Cai, 2018). The key for the success is the use of data augmentation which reduces the classification error rates and issues related to overfitting seen on artificial

neural networks (Bianco et al., 2019). Research of Nanni *et al.* (Nanni et al., 2020) shows that the classification algorithm using traditional methods requires appropriate inputs features from audio spectrograms and they can greatly improve the performance accuracy of the classification model.

Traditional classifiers such as Gaussian mixture models (GMM), hidden Markov models (HMM), random forest (RF), k-nearest neighbor (KNN), decision tree, and support vector machine use a separate task to find inputs features to the classifier (Potamitis, 2014). The term handcraft is used to refer to a manual process of finding optimized inputs feature extraction used by traditional methods of classification and most of these methods are based on time-frequency domain (Nanni et al., 2020; Mcloughlin et al., 2019). However, converting audio traces into their visual representations enabled the use of feature extraction techniques commonly used for image classification. GMM and HMM are based on prototype pattern matching from sample labeled data in its classification (Potamitis, 2014).

The automatic method of data acquisition currently used; will generate a large amount of data when the recorder is used for a long time. Hence further algorithms need to be used to speed up processing, analysis, and classification over a big amount of data. Audio signal spectrograms and Mel-frequency Cepstral Coefficients spectrograms are the fundamental algorithms of these handcrafted features development. In the real sense of what is happening in practice in certain cases; bioacoustics data are relatively small in size and the training phase of deep neural networks requires a large amount of data in the training phase (Nanni et al., 2020). Thus having a small sample size during the training phase of deep neural networks can result in overfitting (jian Xie et al., 2018).

Deep learning methods do both; learning an appropriate number of features based on the depth of the feature extractor automatically and they also perform model classification on species in the moment of training phase (jian Xie et al., 2018). The research of Michael *et al.* (Bianco et al., 2019) shows that the classification performs well with self-learned features in recent years' studies.

The study of Taye (Taye, Hwang, & Lim, 2020) shows that support vector machine (SVM) did not generalize well on certain datasets while performing poorly in the anal-

ysis of bioacoustics data. Syllables of the birds are single notes of the call. Calls have spectrograms images with syllables from various birds species overlapping each other. These call notes do not follow separate linear classes.

Deep learning is able to make a classification with self-learning features and also giving good accuracy compared to other traditional classification models (Taye et al., 2020). This study shows that CNN achieves the highest performance metrics of sensitivity, specificity, accuracy, and area under the curve when compared to KNN, SVM, and ANN. CNN makes a good performance but, the training data required may be large compared to the small available data. These available data may contain an unequal distribution in the target variable of classes. Unequal distribution (class imbalance) leads to the overfitting of classification prediction analysis.

With a huge amount of training data, machine learning (including deep learning methods) can discover models describing acoustic phenomena which are complex (sounds of animals, human speech, reverberation, birds calls) (Bianco et al., 2019). CNN model achieves good performance with large and sufficient data in the training phase. In certain situations, available data are small or imbalanced in classes. Some techniques are therefore needed to solve these issues.

The purpose of the study is to build a CNN model for the automatic detection of Cape robin-chat (*Cossypha caffra*) and pin-tailed whydah (*Vidua macroura*) at a high level of accuracy without overfitting. The research will investigate data augmentation techniques that have been shown to perform well in computer vision and speech recognition to solve overfitting problems for small training data. Having a classification model; the experiments consist of selecting a random number of samples, apply data augmentation technique, testing and evaluate the model on these samples. Results from the group of samples together with applied augmentation methods are therefore compared.

1.2.1 Problem statement

Conservation, management, and monitoring of species is still hard work to achieve. These activities of ecologists are processed manually (Sankupellay & Konovalov, 2018) and biodiversity assessment is challenging from the data collection step to the interpretation

of results of the analysis. During the collection of data, recorders capture animal's sounds or birds call. The complexity comes to the signal pre-processing and analysis of data.

Although the CNNs model performs well on classification predictions, the data for model training are limited (available data are not enough to train the model) in certain cases. When talking about the limitation on data, this means a lack of data. Samples of at least one class may not have enough representation during model training and this produces an imbalanced dataset. Call from rare species is an example that can create an imbalanced dataset because these species do not call often. Thus, there might be few documented examples of their calls. For example, there are only three recordings publicly available for the endangered Hainan Peacock-Pheasant ¹.

1.2.2 Objectives

Main objective

The primary objective is to apply data augmentation techniques used in other areas of machine learning research, for example speech recognition for limited data and perform an investigation of these techniques on classification prediction using performance metrics such as accuracy. The specific objectives of this thesis are listed below:

Specific objectives

1. To analyze the characteristics of pin-tailed whydah *Vidua macroura* and Cape robin-chat *Cossypha caffra* calls (audio data).
2. To apply appropriate pre-processing techniques on collected audio recordings. The research is considering the techniques that were used for other types of audio data.
3. To construct a public bird song dataset having annotations and calls of pin-tailed whydah and Cape robin-chat.
4. To apply 4 data augmentation techniques on 3 types of calls (pin-tailed whydah, Cape robin-chat and noise). The classification results from these 4 data augmentation techniques are compared with a baseline.

¹Hainan Peacock-Pheasant <https://www.xeno-canto.org/species/Polyplectron-katsumatae>

5. To train a convolutional neural network for classifying pin-tailed and Cape robin-chat species.
6. To evaluate the trained classifiers on test data for each of the 4 augmentation techniques.

1.3 Organization of the research

The entire work of the study is organized into 5 chapters. Chapter 1 provides a general introduction to the problem, chapter 2 discusses relevant literature, the research methodology is presented in chapter 3, the results are revealed in chapter 4, and finally, chapter 5 concludes the thesis and provides recommendations.

Chapter 2 Literature review

2.1 Introduction

Given the lack of literature on augmentation techniques being applied to the field of bioacoustics (Dufourq et al., 2021), this chapter will review studies where augmentation has been used in computer vision tasks. Limited data may be fixed with data augmentation which is increasing the samples size by creating new synthetic data from a small amount of existing samples (Dufourq et al., 2021; Taye et al., 2020; Mikołajczyk & Grochowski, 2018) and there are a number of techniques of data augmentation using traditional methods or based on computer vision. Augmentation techniques used in the above studies were geometric transformation (rotation, cropping), generative adversarial network (GAN), image finishing style (image on-blur), random erasing (on a small region of the image), noise addition, time-shifting, a mixture of images (blending) and the use of the public repository. Results from data augmentation in prediction classification models have been good in performance. Data augmentation is not new, it has been used for computer vision (image and video classification) studies as well as other researches related to machine learning and deep learning (Perez & Wang, 2017; Mikołajczyk & Grochowski, 2018; J. Wei & Zou, 2019; Jiao, Tu, Berisha, & Liss, 2018).

This chapter provides a brief overview of the data augmentation term, its usage, techniques involved, related work, and performance achieved in applying some of the augmentation techniques. Performance is indicated by visualization plot, error rate, or with other metrics values such as sensitivity, precision, recall, accuracy, and f1-score.

A good prediction model is achieved by creating additional artificial samples from original existing data (creating many more new samples of data) (Dufourq et al., 2021). This increases the size of training data and therefore it fixes overfitting issues while improving the accuracy of classification predictive model (Perez & Wang, 2017). Geometric

transformation (cropping and zooming) and changing colors are common approaches of data augmentation (Perez & Wang, 2017) for image classification.

The above examples of the literature clearly show that augmentation defines a set of methods used to provide new data points from existing data. This approach has a powerful application in machine learning, particularly in deep learning since the cost for re-collecting data and, the effort of labeling them for classification are catted off. Augmentation fixes issues related to the class imbalance of the classification, and limited data while improving model performance and accuracy. Many examples relate to computer vision and few of them relate to bioacoustics. The next paragraph explains why all available augmentation techniques could not be feasible for the bioacoustics dataset.

Data augmentation methods like time shifting (Dufourq et al., 2021), Gaussian white noise injection, random multiply, image blur, vertical and horizontal roll, random crop, dropout, blackout approaches (Koh et al., 2019) have been used for augmentation with a similar purpose to improve the classification performance of computer vision studies. Not all of these techniques are appropriate for bioacoustics because of the temporal nature of the special physical (spectrum) and the sounds of the signal (S. Wei, Zou, Liao, et al., 2020). Augmentation using transformations of data by adding Gaussian noise, time stretch, and pitch shift is commonly used for audio augmentation because time-frequency domain constraints of the signal maintain the label of the original sample. Audio augmentation relies on the simple concept of mixing up two samples of the same class and other data augmentation methods for the reconstruction of the labels in image processing. The class label of the original sample is preserved on the newly created sample by using augmentation.

2.2 Related work

The study of Perez *et al.* (Perez & Wang, 2017) shows that three augmentation techniques namely traditional transformations, GAN, and learning augmentation have been used to classify dogs versus goldfish, dogs versus cat, and MNIST 0's and 8's. The traditional transformation was based on the geometric concept of the shifted image, zoomed in/out the image, rotated image, flipped image, distorted image, or apply the shade with a hue on the image.

The study of Luis *et al.* (Perez & Wang, 2017) shows that Cezanne, enhance, Monet, ukiyoe, van Gogh, winter are 6 styles of GAN augmentation to generate 6 new images. “*Learning the augmentation*” is the term used in the above study to refer to the augmentation technique which combines two different images of the same type into a single image as output. “*Neural networks with no loss*” is a title given to an augmentation neural network with a no loss approach and it had got 91.5% as the highest accuracy in classifying dogs versus goldfish (accuracy of 85.5% was achieved with no augmentation). The same augmentation had got 77.5% as the highest accuracy in classifying dogs versus cats (accuracy of 70.5% was achieved with no augmentation). The augmentation best accuracy in all used datasets was given by MNIST in classifying 0s and 8s. This accuracy was 97.2% without augmentation methods and it reached 97.5% with augmentation. The change in the accuracy is 0.3 for MNIST, the change in accuracy is 6% for dogs versus goldfish classification and 7% for dogs versus cat classification. The last two predictions have a good improvement in accuracy but they are not the best score. MNIST has less change in accuracy but overall accuracy is excellent compared to the other two remaining datasets.

The study of Nanni *et al.* (Nanni et al., 2020) shows that four techniques have been used and three of them improved the performance of the model classifier when compared with no use of augmentation approaches. Accuracy on spectrogram augmentation method using fusion-local on cat dataset has got the highest accuracy of 91.73% compare to other used augmentation techniques for all neural network architecture (VGG16, GoogleNet, VGG19 and etc). “*Fusion - Local*” is one type of neural network architecture used for this study.

Accuracy on “*baseline*” which is a no augmentation (NoAUG) method using “*Fusion Si+Sp+SSG*” (a CNN) on BIRDZ (audio of bird species from xeno-canto website) dataset. Xeno-canto¹ is a public worldwide site for sharing bird sounds. The baseline has got the highest accuracy of 96.85% compared to other used augmentation techniques. Standard image augmentation works on basis of geometric transformation (reflection, rotation) and computer vision. Standard signal augmentation results in 10 more copies

¹xeno-canto <https://www.xeno-canto.org/>

of original audio using built-in augmentation methods of MATLAB (speeding up audio, time, and pitch shift, adding noise and volume increase). Spectrogram augmentation also creates six versions of new copies of the spectrogram from the original one. The last is signal augmentation which is working directly on raw audio. All these augmentation techniques were used in the referenced study. This study also describes the CNNs architectures used. These are GoogleNet, VGG16, VGG19, GoogleNet – places365, VGG16 - batchSize, and GoogleGoogle365 (sum rule of GoogleNet and GoogleNet-places 365 trained with each of the data augmentation protocols) convolutional neural networks architectures. GoogLeNet was developed by Google and it is a convolutional neural network with 22 layers in deep. GoogleNet was trained with “*ImageNet*” dataset² having 1000 objects categories. GoogleNet-places 365 is similar to GoogLeNet except that it classifies images into 365 different place categories (field, park, runway, and lobby). These network architectures have been used with/without augmentation techniques for birds and cats audio data. GoogleNet got 82.98% without augmentation, 76.44% using “*standard image augmentation*” approach, 85.12% on “standard signal” augmentation method, 85.25% on “signal augmentation” method and finally 88.68% by using “*spectrogram*” augmentation technique.

VGG16 got 84.07% without augmentation, 77.02% using standard image augmenter, 86.64% on standard signal, 88.20% on signal augmentation and finally 90.71% by using spectrogram augmentation. The performance using standard image augmentation is lower comparing to other augmentation techniques. This performance is still less when it is also compared to the use of the no augmentation approach. Spectrogram augmentation outperforms well for all augmentation techniques used of this study. The three remaining augmentation techniques have only improved accuracy.

The study of Mario (Lasseck, 2018) shows that a number of 1500 birds species (a large collection of audio recordings provided by Xeno-Canto) were classified using data augmentation. Birds call are converted to spectrogram images. These spectrograms were again segmented into birds only, noise only, background atmosphere, and low quality or highly compressed recordings data sets. The study has used a number of data augmentations techniques but the full augmentation was the best compared to other methods. Here are some augmentation methods used such as combining two different files from

²ImageNet data <https://image-net.org/download.php>

different datasets, extracting parts from a random position in the file, adding two files together, applying jitter duration and etc. Jitter duration is an augmentation method to add random brightness, contrast, saturation, and hue to the existing image to form a new copy of data.

The performance is measured using mean reciprocal rank in percentage (MRR) and it achieved 65.538% without augmentation and 74.466% with full data augmentation (i.e combination of augmentation techniques). MRR in full augmentation except with *NoiseOnly* (training file having audio content without signal) and *AtmosOnly* (combining longer sequences of successive frames related to noise with an overall duration greater than one second) had 67.893% which is presenting a slight change compared to the use without data augmentation. Except noiseOnly and AtmosOnly, all other combinations achieved the same results as the full augmentation method. The study experiment shows that the use of noiseOnly and AtmosOnly has a major improvement to the performance. MRR improves 10% in the classification by using full augmentation method, i.e a complete augmentation got 74.4% while methods without data augmentation are having 65.538%.

The study of Xiaofeng *et al.* (X. Zhang, Wang, Liu, & Ling, 2019) shows that two data augmentation approaches called traditional data augmentation (C-augmented) and deep adversarial data (DADA) were used. As the number of samples increases the accuracy of the classifier achieved on limited data known as C is the lowest; DADA only improves in the accuracy compared to C and DADA applied on C-augmented. DADA-augmented has the highest accuracy in the results. Again DADA-augmented had made 65.49% from 59.9% and 62.7% using f1-score. Mean fold model (MF), long short-term memory (LSTM), a digital database for screening mammography (DDSM), and DS specify the impact of transformation function domain-specific. The last two of the above metrics are the f1-score of MF and LSTM tasks to increase performance used for DDSM plus DS. All these performances are found in (Ratner, Ehrenberg, Hussain, Dunnmon, & Ré, 2017) study.

The study of Alexander *et al.* (Ratner et al., 2017) shows that the basic, heuristic, mean fold model (MF), and long short-term memory (LSTM) are the main data augmentation used for this study. The datasets used for the experiments are CIFAR-10,

MNIST, and ACE. The first two datasets are well known in the other studies of this literature review chapter. MF and LSTM augmentation techniques are also well known in this chapter. Unlike CIFAR-10 and MNIST datasets; ACE is a different dataset. Automatic content extraction (ACE) is a text dataset (corpus) and this technique is used to extract relation in the text of the sentence. Performance is measured using accuracy for the CIFAR-10 and MNIST datasets, and f1-score is used for the ACE text dataset.

The test with 1% of MNIST dataset got 96.7 f1-score using LSTM (90.2 without augmentation) and 10% on MF with 99.2 f1-score (97.3 without augmentation). Test with 10% of CIFAR-10 dataset got 81.5 f1-score using LSTM (66.0 without augmentation) and 100 percent on MF with 94.4 f1-score (87.8 without augmentation). ACE using f1-score got 64.2 using LSTM (62.7 without augmentation). These are the best performance measurements from augmentation compared to the use of no augmentation. Basic and heuristic augmentation techniques have improved in performance but did not produce the best overall score.

The study of Jia *et al.* (Shijie, Ping, Peiyi, & Siping, 2017) shows that GAN, Wasserstein GAN (WGAN), flipping, cropping, shifting, PCA and color jittering, noise, rotation, and some of their combinations as data augmentation techniques. Results show that there are improvements in performance for individual data augmentation except for the addition of noise. Their combination shows a significant change of improvement from the use without/with augmentation. The test was done using the original dataset only, the original dataset with data augmentation, and finally initial data together with a double of augmentation data. ImageNet and CIFAR10 datasets were used to test WGAN, cropping, rotation, flipping data augmentation, and their combination pairs. Results on the individual augmentation approach with the data size of triple (original training set plus the double size of the generated samples) are highest. Pairs of augmentation methods also achieved the highest results in a similar way to the individual techniques of augmentation.

The study of Mingyang *et al.* (Geng, Xu, Ding, Wang, & Zhang, 2018) shows that augmented random search (ARS-Aug) was used to improve the existing AutoAugment approach. PyramidNet + ShakeDrop achieves a model small error rate of 1.26% on testing CIFAR10. AutoAugment had 1.48%. On ImageNet test errors are 10.24% on ARS-Aug whereas AutoAugment is 10.67%. Small errors mean the better the model is.

Validation set Top-1 / Top-5 error rates are AutoAugmenter: 16.46/3.52 and ARS-Aug: 16.12/3.28 using AmoebaNet-C (6,228) model. Errors are reducing which gives a clear indication of the improvement.

The study of Philip *et al.* (Jackson, Atapour-Abarghouei, Bonner, Breckon, & Obara, 2019) shows that Color jitter and style Augmentation (uses a random texture, contrast, and color, while keeping shape and semantic content) techniques were used as data augmentations. InceptionV3 achieved the best prediction accuracy of 0.765, 0.893, and 0.215 on three office datasets using style augmentation.

Spectrograms and images are two dimensions visual objects, therefore the image can be treated like spectrograms and vice versa. There is a possibility to perform similar operations on both of them. Spectrograms represent the strength of the signal using the time-frequency domain while images represent pixels using width-height axes. Width turn to the time axis, height becomes frequency and signal strength turns to the pixels. So the analysis of images in terms of computer vision is applied completely to audio signal processing with spectrograms.

The study of Wong *et al.* (Wong, Gatt, Stamatescu, & McDonnell, 2016) shows that two types of data augmentation (data space and feature space) were used. Data space; image data creates new copies by transforming existing samples. New samples preserves label information with validation of label integrity being performed by a human observer. Feature space; image creates new copies of data by an arbitrary transformation of an existing sample by also preserving label information. The result takes baseline performance (CNN, CSVM, and CELM) on MNIST data. Each of the techniques used in the baseline are tested with real data, Synthetic Minority Over-Sampling Technique (SMOTE), elastic distortions to images of existing samples and Density-Based Synthetic Minority Over-Sampling Technique (DBSMOTE). Overfitting is indicated by the graph of the results, i.e when the graph shows the gap space between training errors and testing errors is big.

On the baseline, the CNN reduces the overfitting as the number of samples increases. On CSVM, the DBSMOTE increases overfitting to its highest value. It produces poor performance. On CELM, increasing real samples data shows that the overfitting gets reduced.

The study of Terrance and Graham (Devries & Taylor, 2017) shows that the used augmentation method is similar to the above study of Wong *et al.* Results show that the testing errors in percentage keep on reducing while samples increases. Performance is good when testing errors are reducing. The study by Agnieszka and Michał (Mikołajczyk & Grochowski, 2018) shows that the traditional image transformation geometric and color, GAN, style transfer, finishing, and their combination with GAN. Results show that the combination of augmentation techniques provides a significant improvement on accuracy.

The study of Zhun *et al.* (Zhong, Zheng, Kang, Li, & Yang, 2020) shows that ResNet-101 model's errors reduced to 5.30 from 5.73 (baseline) on top-5 and 20.43 from 20.98 (baseline) on top-1 on ImageNet. Comparing dropout, random erasing, and baseline; random erasing achieved a small test error rate of 4.31 while it was around 5 error rate for others. Comparing random flipping, random cropping, and random erasing with baseline; again random erasing has a smaller error rate i.e model improves. Results from this study show a small error rate compared to baseline errors.

In summary, the literature reveals that data augmentation improves the performance of machine learning classifiers. The use of individual augmentation techniques results in minor performance improvements, whereas combinations of techniques result in significant improvements. Furthermore, the literature reveals that convolutional neural networks are currently state-of-the-art for creating acoustic classifiers. Based on these findings we explore data augmentation techniques for bioacoustics classification in the context of small datasets which is relevant when monitoring endangered species. To achieve this we will apply data augmentation techniques to the dataset which we have manually annotated. The description of these details is found in the next chapter.

Chapter 3 Methodology

In this chapter, we first discuss how the data was collected, annotated, and provide a description of data pre-processing. Finally, we present our proposed methodology and related terms used for the implementation of the research and data augmentation to be used for the research.

3.1 Data Collection

The data for this research consists of audio recordings taken from Intaka Island Nature Reserve, Cape Town, South Africa (Figure 0.1). The recorders were set to record over a period of two weeks from 3:00 am to 7:00 pm. The audio recordings of birds are divided into training and testing sets. One week of data was used as training data and the other week was used as testing data.



Figure 0.1: Intaka Island Nature Reserve, Cape Town, South Africa.

All recordings were obtained using an Audiomoth (Hill, Prince, Snaddon, Doncaster, & Rogers, 2019). The calibration of the sampling rate is set to 48,000Hz. An audio file is stored on the computer in the form of a digital signal. The digital signal uses the numbers to represent amplitudes within the sound signal. When a computer is reading the file, it needs to know the number of amplitudes to execute in one unit of time (i.e number of amplitudes per one second). This number of amplitudes is known as sampling rate and it is measured in hertz (e.g. 48,000Hz). For instance, a sampling rate of 48,000Hz indicates that the computer is reading an audio file by executing 48,000 amplitudes at once (48,000 amplitudes values in one second). The sampling rate is the number of audio amplitudes/points that the recording device can record in one second (e.g. 16,000 data points per second).

The sampling rate may be set using different calibrations (8000Hz, 16000Hz, 32000Hz, 48000Hz and etc). By selecting a high-frequency value on the recorder, results in a high range of frequencies that can be recorded hence more species are captured. With this fact in mind, the sampling rate of 48000Hz will capture more species from the island to help further/ future research. Birds species of interest in the research is the pin-tailed whydah (*Vidua macroura*) in figure 0.2 and Cape robin-chat (*Cossypha caffra*) in figure 0.3. These birds are common species in South Africa and it is easy to collect their call.



Figure 0.2: Adult male pin-tailed whydah, extracted from eBird (*Pin-tailed Whydah - eBird*, n.d.).



Figure 0.3: Cape robin-chat, extracted from eBird (*Cape Robin-Chat - eBird*, n.d.).

An AudioMoth (Figure 0.4) is a low-cost, open-source acoustic monitoring device used for monitoring wildlife. It is not only sensitive to audible sounds but well into the ultrasonic frequency range (Hill et al., 2019). It records uncompressed audio from 8000 up to 384,000 samples per second. AudioMoth has lower cost, lower power utilization, small size and it is easy to use when compared to existing recorders such as SongMeter series.



Figure 0.4: An attachment of AudioMoth recorder to the tree.

Analysis of the calls in frequencies finds that the pin-tailed whydah call has a maximum frequency of 7-8KHz while the call for Cape robin-chat has a frequency of around 3KHz. Nyquist theorem (Shannon, 1949) states that the signal is regenerated without loss of information if the sampling frequency is set at least double of the maximum frequency in digital signal processing. If considering 3KHz frequency, the Nyquist frequency will be 6KHz which is enough to generate a digital signal of Cape robin-chat. By doing so the pin-tailed whydah will however result in a big loss of information in its digital signal. Taking the 8KHz frequency of the pin-tailed whydah will satisfy the generation of a digital signal on both species by the Nyquist theorem (i.e 16KHz). The frequency of 16KHz will create an excellent digital signal for Cape robin-chat and pin-tailed whydah as well. So the maximum frequency for both species is 8KHz, and it is good practice to set the frequency at 16KHz or higher for capturing the species being surveyed.

3.1.1 Pre-processing

Image classifier CNNs models usually require fixed sizes as inputs. One can apply a 1D CNN to raw amplitudes but the study did not use this approach since the literature (jian Xie et al., 2018; Mesaros et al., 2017) shows that better performance can be achieved when converting the amplitudes to spectrograms. Audio data are converted to spectro-

grams images that can be input using 2D CNNs architecture. SonicVisualiser¹ is free and open-source software designed generally for visualizing, analysis, and annotation of audio recordings. It reads audio in form of waveform (Figure 0.6).

The fact of setting up the recorder at 48KHz makes it capture many species including pin-tailed whydah and Cape robin-chat calls. This digital signal has additional calls apart from the two main calls of the study (pin-tailed whydah and Cape robin-chat). This is an advantage for further research to use the same data but it is a challenge in dealing with the pin-tailed whydah and Cape robin-chat calls. The use of a filter solves this issue by only selecting signal samples/amplitudes with frequencies involving the main call of the study. The filter removes signals from above 16KHz to 48KHz. Choosing 16KHz is due to the pin-tailed whydah call frequency of about 8KHz for the analog signal which is 16KHz by Nyquist theorem in the digital signal processing. The filtered signal may introduce artifacts which are therefore fixed by the down-sampling stage.

Down-sampling is a technique used to reduce the signal sample rate. Sample rate in digital signal processing defines the number of samples taken from the signal in one second. The first pre-processing step is to filter to a certain frequency to avoid artifacts that can be caused by down-sampling. The filter isolates a certain range of frequencies from a spectrum of frequencies. Filters select a part/band of frequencies from the spectrum depending on the needs. A low pass filter, high pass filter, or band-pass are types of filters. The signal is filtered in the range 2500Hz to 7000Hz because the frequency band of the species being surveyed falls into that range and the practical frequency on both species is 16KHz by Nyquist theorem. Down-sample stage is the next stage as explained in the previous paragraph.

The spectrogram (example Figure 0.5) visual representations are annotated by annotating the start and end time of an event of interest (call or noise). The start and the end of these events are indicated by boxes depending on what is being annotated. Short vocalization events will have shorter annotation boxes compared to longer vocalization events. The CNNs require fixed input images. To achieve this, one can extract fixed

¹SonicVisualiser <https://www.sonicvisualiser.org/>

audio segments of some pre-defined duration. Then the extracted audio is converted into a spectrogram. Using this manner, one could create a dataset of spectrograms for which each is of the same size. The pin-tailed whydah calls are much longer than 3 seconds. Choosing a large segment would contain a lot of information and sometimes the Cape robin-chat calls were short (e.g. 1 to 2 seconds). Thus 3 seconds is a window size that ensures that at least the Cape robin-chat call is captured. If having a longer annotation box (e.g. 20 seconds annotation), the extraction of multiple 3 seconds segments will be 18 segments ($18 = 20 - 3 + 1$) of 3 seconds each. These segments are taken by jumping 1 second from the starting point. If the annotation time in seconds is less than 3, only one segment is extracted. Here there is a general formula to compute the number of segments to extract.

$$n = t - 3 + 1 \quad (0.1)$$

Where,

n denotes the number of segments,

t denotes the time duration in seconds of annotation ($t \geq 3sec$).

These fixed inputs segments of 3 seconds are applied to the CNNs model. Short-time Fourier transform and digital signal processing concepts are applied in the pre-processing phase.

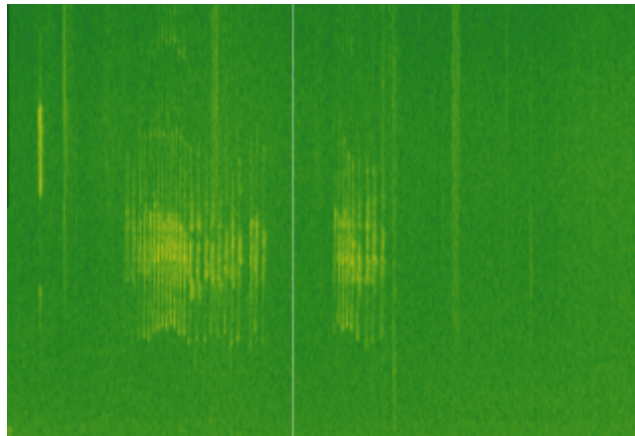


Figure 0.5: An illustration of a spectrogram image.

An example of spectrogram of the audio signal (Figure 0.5).

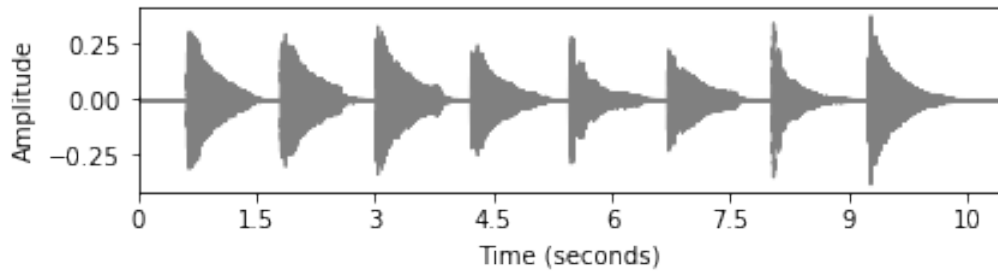


Figure 0.6: An illustration of a waveform image.

Above figure 0.6 has a waveform showing changes of signal amplitudes over time while spectrograms display changes of frequencies over time.

Figures 0.7 and 0.8 show examples of spectrograms for the call of the Cape robin-chat. The annotation CRC denotes the labeling that was used in this study for the call of a Cape robin-chat. The value of 10 denotes the annotators' confidence. A value of 10 meant the annotator was confident, a value of 1 annotated that the annotator was not confident. This allowed for the creation of a dataset whereby varying levels of confidence were assigned and could then be verified by an expert. The 768 audio files from two weeks of data collection (one week for testing and one week for training) resulted in 256 hours. The 256 hours of recordings were manually verified in this manner and bounding boxes were created. The start and end time of each box can be obtained and thus enables the creating of a dataset.

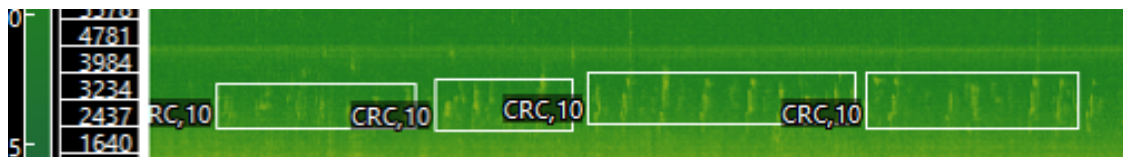


Figure 0.7: Sonic visualizer annotation images.

Annotation of Cape robin-chat call indicated by rectangle boxes of varying length (Figure 0.7).

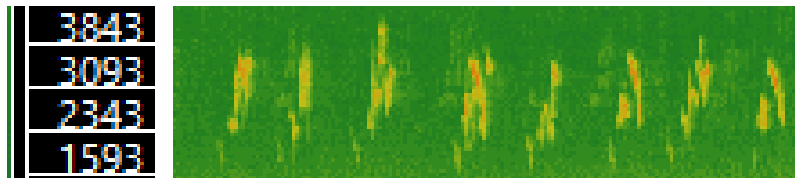


Figure 0.8: Another call of Cape robin-chat call image

Figure 0.8 shows that the call of Cape robin-chat is ranging from 2300Hz to 3000Hz on frequency axes.

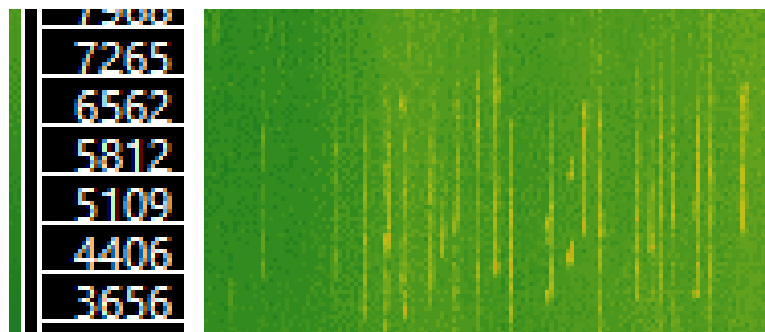


Figure 0.9: Example of pin-tailed whydah call images.

Figure 0.9 shows that the pin-tailed whydah is ranging from 3000Hz to 7000Hz on frequency axes.

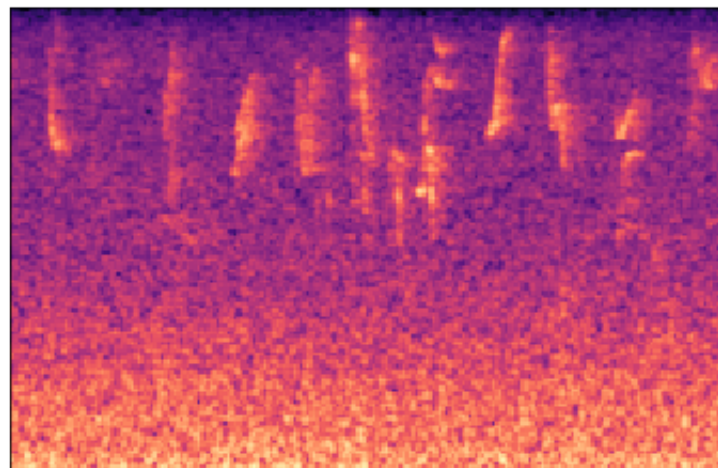


Figure 0.10: Typical input image of pin-tailed whydah call to CNNs model

Figure 0.10 shows original call of pin-tailed whydah that can be an input to CNNs model. This spectrogram has a height of 126 and a width of 214. This shape (126,214) is 3 seconds fixed in time. This shape was initially varying in length.

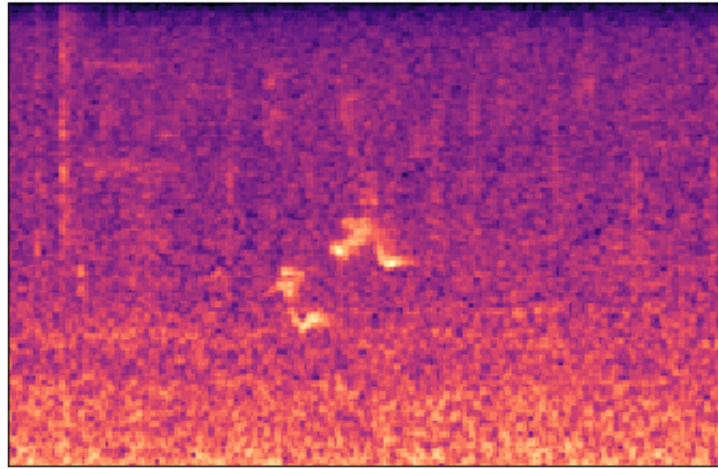


Figure 0.11: Typical input image of Cape robin-chat call to CNNs model

The following tasks are conducted in pre-processing stage:

1. Sonic Visualiser, an open-source software, was used to to annotate individual species of our interest by listening to sound and drawing a box to the call of species being surveyed. There are several applications and tools to analyze audio music files but Sonic visualizer is used since it is free and open-source software and, EdgeAcoustic organization has provided training of using this tool for bioacoustic data analysis and making call annotations.
2. From audio and annotations; SVL (Scramdisk volume file) contains metadata (start and end of the call annotated) is created from sound and annotations and it has the form of the data frame.
3. Extracting individual sounds of the pin-tailed whydah, Cape robin-chat and noise (calls of other species or background sound) from recordings based on metadata file. Metadata contains basic information about data and this information is used to access or manipulate the data.
4. Apply low pass filter and down-sampling. A default sampling rate is 22050Hz and based on species being surveyed which has a maximum frequency of around 8Khz,

this makes down-sampling to be at 16Khz (a sample rate of 16000 samples/second) of the audio signal according to Nyquist theorem.

5. Convert these audio waves to the spectrogram using a short Fourier transform.
6. Extract segments of fixed length using spectrograms and annotations.

3.1.2 Data after the pre-processing phase

The 768 audio files have been pre-processed where each audio file has 20 minutes. Therefore, the 768 audio files resulted in 256 hours. These audio files are the (.wav) audio file extension. The annotations were made for all the 768 audio files, 15 hours of recordings were made publicly available along with their corresponding annotations. The annotations are in the (.svl) files and both files (.wav and .svl) can be opened using Sonic Visualizer. The public dataset² is accessible from the zenodo website.

3.2 Machine Learning

Analysis of bioacoustics data requires a human being to spend time and a lot of effort listening to audio calls and predict the labels of species being involved. Automatic operations are therefore introduced for analysis with the help of a computer machine. It is important to analyze, detection and classify bird species with automatic operations.

Machine learning is a data-driven technique and it represents methods/algorithms of automating data processing and pattern recognition (Bianco et al., 2019). These algorithms are built based on statistics (Jiao & Du, 2016). Statistics is used to support machine learning algorithms due to uncertainties within data. Machine learning takes available data and divides them into three datasets, one for training model, validation to evaluate the model, and testing to test the final model with unseen data (Jiao & Du, 2016).

Machine learning is divided into two main categories, supervised learning (input and output labels are given in advance) and unsupervised learning where labels are not

²Pin-tailed whydah and Cape robin-chat calls for passive acoustic monitoring <https://zenodo.org/record/5141676#.YQvPtugzbIV>

given (Bianco et al., 2019). Supervised learning is further divided into two classes namely regression (numerical continuous output variable) and classification for categorical output variable with two or more labels in the target variable (Bianco et al., 2019; Bermant, Bronstein, Wood, Gero, & Gruber, 2019).

Analysis of prediction classification of bioacoustics data with traditional methods models requires extracting input features by handcraft methods (Nanni et al., 2020). After the computation of these features, a classification algorithm is applied to predict the label. High dimensional input features are also an issue for the general machine learning algorithm, therefore, a feature reduction algorithm is used to fix this problem. Deep learning is more flexible since it can learn filters that produce feature maps automatically.

3.3 Deep learning

Deep learning networks are computational models with multiple layers of processing units for learning representations of data at many levels of abstraction (LeCun, Bengio, & Hinton, 2015). Methods in deep learning have made a significant improvement in computer vision (Shijie et al., 2017), speech recognition (Dufourq et al., 2021), object detection (Perez & Wang, 2017), and many other systems (LeCun et al., 2015).

Deep learning is a subset of machine learning techniques that can learn filters to produce automatically feature maps from actual input data during the training phase of the predictive model (jian Xie et al., 2018; Bermant et al., 2019). The term deep learning is derived from the use of many layers and functions from the input side of the model to its output (Bermant et al., 2019). Deep learning refers to multi-layer neural networks such as recurrent neural networks (RNN), CNNs, and many more derived from these architectures (Shrestha & Mahmood, 2019).

3.3.1 Artificial Neural Networks

An artificial neural network (ANN) (Bre, Gimenez, & Fachinotti, 2018) is a non-linear statistical modeling method, which is biological in nature. Figure 0.12 illustrates an example of an ANN with 4 input nodes (in the input layer), 2 hidden layers, and 1 output nodes (in the output layer). Each node is fully connected which means that

every node of the previous layer has a direct link to the node of the next layer. Statistical models are parametric, they rely on some assumptions on data. Unlike statistical models, ANN is non-parametric models (Abiodun et al., 2019). ANN forms a neural network from the input layer to the output layer (Cormack, 1971; Abiodun et al., 2019). Inputs layer has nodes (neurons) that receive input features and they do not include any processing ability. The output layer is associated with the target variable. One or more hidden layers are placed between the input and output layer and links or connections between nodes define weights values.

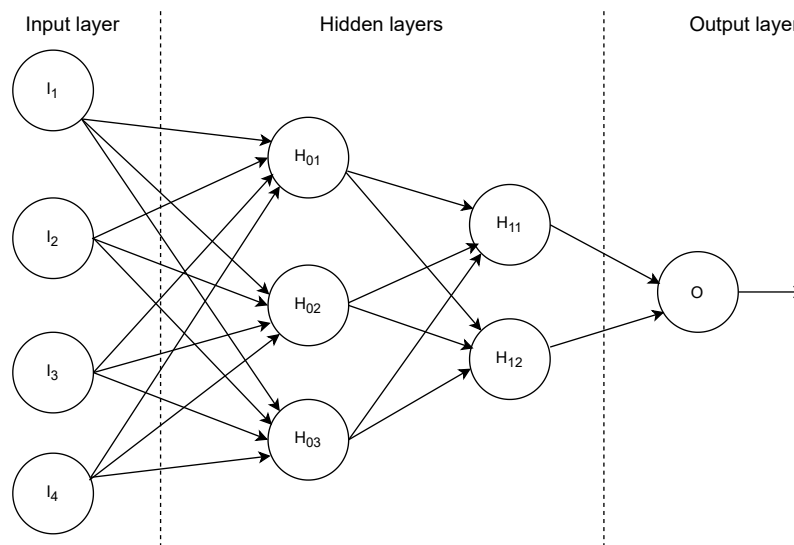


Figure 0.12: ANN with one input layer, two hidden layers and one output layer.

Nodes with processing abilities perform two main operations inside (summation and activation). The summation operation takes inputs together with their weights by summing up them. These inputs may be the output from the previous layer. Depending on the nature of the problem ANN may solve regression problems as well as classification. The most and successful application of ANN is pattern recognition (Abiodun et al., 2019).

Bias is an offset, it helps in shifting the activation function by adding a constant value and it will help the model to produce a good fit to the data. Initial weights are randomly created and the total loss between real and predicted values on target are computed during training. The training backpropagation process updates initial weights

until errors are minimized.

Gradient descent is an iterative process and, an optimization method used to improve deep learning neural networks performance metrics by minimizing the cost function (errors are minimized during the back-propagation of ANN) (Abiodun et al., 2019). Gradient descent finds the minima of the function. The gradient descent technique provides parameters (learned weights) that are optimal if the function is at the global minima. So this method does not guarantee to find the global minima since it may stop at the local minima without reaching the global minima point. Feedforward neural network or multi-layer perceptron (MLP) is composed of one input layer, one output layer, and at least one hidden layer and it is the most applied ANN architecture.

3.3.2 Activation and loss functions

An activation function is a function that applies some mathematical function on whether a neuron should be activated or not by computing a weighted sum, and bias is added in addition to the weighted sum. Without a non-linear activation function, a neural network is a linear model. So the non-linear activation is added to introduce a non-linear (Feng & Lu, 2019) transformation on input making it to be able to perform and learn tasks that are more complex. There are two main categories of activation namely linear and nonlinear function (e.g.Sigmoid).

Linear function

This function (Equation 0.2) forms an algebraic equation whose graph is a straight line and its output changes are proportional to the inputs.

$$f(x) = k * X \tag{0.2}$$

x is an input variable to a linear function. f(x) is the corresponding output. k is a constant. The derivative (Equation 0.3) gives:

$$f'(x) = k \tag{0.3}$$

which is showing that the error cannot be decreased further by the gradient. The gradient is fixed to a constant.

Sigmoid Function

The graph of this function (Equation 0.4) forms an S-Shape, the range of its output is $[0,1]$ and it is nonlinear by nature. This activation is popular for classification problems where the computed probability is at least 0.5 the output changes to 1 and 0 otherwise. It has the form of :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (0.4)$$

and its derivative (Equation 0.5) is:

$$f'(x) = \frac{e^{-x}}{(1 + e^x)^2} \quad (0.5)$$

One needs to pay attention to initializing the weights of sigmoid due to the vanishing problem. This problem occurs when the gradient is too small almost closer to zero which results in no change for the new weight from the old weight. It provides a slight decrease and prevents the training to learn optimized parameter values.

Softmax Function

Softmax (Equation 0.6) is for problems with multivariate classification where models working on multi-class; return probabilities of each class and the target class will have the highest probability among others (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The sum of probabilities associated with classes in prediction must be equivalent to 1. This function is mostly used in almost all the final layers of the deep learning neuron networks once they used. The experiment will use this type of function as it is a non linear function, it is mostly used for final layer of the classification and a 3 label classification of the research is part of multi-class task. The softmax is given by the following mathematical relationship:

$$f(x_i) = \left(\frac{e^{x_i}}{\sum_j e^{x_j}} \right) j = 1, \dots, n \quad (0.6)$$

Hyperbolic tangent Function

Hyperbolic tangent (Tanh) activation function (Equation 0.7) or Tangent Hyperbolic Function is derived from sigmoid as follows:

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 = 2 * \text{sigmoid}(2x) - 1 \quad (0.7)$$

Its derivative (Equation 0.8) is

$$\tanh'(x) = \frac{4e^{-2x}}{(1 + e^{-2x})^2} \quad (0.8)$$

This activation function does not suffer from vanishing problems and it is a non-linear function. Its bound changes between [-1,1] with zero at the center on output.

ReLU

This function (Equation 0.9) is popular for ANN hidden layers, deep learning networks for the last few years.

$$f(x) = \max(0, x) \quad (0.9)$$

It is a non-linear function since the output is always zero for negative numbers and it uses less computation which is making it to be faster than *tanh* and sigmoid activation functions that were popular activation functions for deep neural network (Szandała, 2021). The gradient is zero for no positive inputs, weights will not be adjusted and not responding to inputs variation. This inactive state is making it to be called “dying ReLU”.

Leaky ReLU

Leaky ReLU (Equation 0.10) is a similar function to ReLU except that it has a small slope for negative values instead of a flat slope (Feng & Lu, 2019).

$$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (0.10)$$

Its derivative (Equation 0.11):

$$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \quad (0.11)$$

Coefficient α represents i^{th} channel of no negative input and it is usually small around 0.01. It improves the ReLU activation function for the problem of “Dying ReLU”.

PReLU

This activation function (Equation 0.12) is similar to ReLU except that the parameter α of different channels is learned by networks during back-propagation (Feng & Lu, 2019). For $\alpha = 0$ PReLU becomes ReLU and changes to Leaky ReLU in case of α small and fixed.

$$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (0.12)$$

Derivative is given by (Equation 0.13):

$$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \quad (0.13)$$

RReLU

This activation function (Equation 0.14) also improves the coefficients over negatives inputs like Leaky ReLU, PReLU, and RReLU (Feng & Lu, 2019). The random negative slope is introduced for overfitting purposes.

$$f(x_{ji}) = \begin{cases} \alpha_{ji} x_{ji} & \text{for } x_{ji} < 0 \\ x_{ji} & \text{for } x_{ji} \geq 0 \end{cases} \quad (0.14)$$

where

$$\alpha_{ji} \sim U(l, u)$$

$$l < u; l, u \in [0, 1)$$

α_{ji} is a random number from a uniform distribution bounded by l and u . Integer i refers to the channel whereas j refers to the example.

ELU

Exponential Linear Unit (ELU on equation 0.15) works on the basis of ReLU activation function by speeding up the learning and fixing the vanishing gradient problem (Feng & Lu, 2019).

$$f(x_i) = \begin{cases} \alpha_i(e^{x_i} - 1) & \text{for } x_i \leq 0 \\ x_i & \text{for } x_i > 0 \end{cases} \quad (0.15)$$

Derivative function (Equation 0.16) is

$$f'(x_i) = \begin{cases} f(x_i) + \alpha_i & \text{for } x_i \leq 0 \\ 1 & \text{for } x_i > 0 \end{cases} \quad (0.16)$$

Loss functions

Loss function defines the deviation between the predicted value and target value. Squared-error loss is a very common loss function used for linear regression, calculation of unbiased statistics, and many fields of machine learning (Pilon, 2015). It is used to evaluate the algorithm used for modeling data. Predictive models behave well if the loss function has few errors. If the loss function has more errors the model becomes unusable and will have a poor accuracy. The loss function is a good indicator to guide a potential improvement to be done from the existing model. The loss function also called error or cost function measures how bad or good the model is in terms of performance. Here the goal is to find a mechanism to minimize errors for regression and classification tasks as much as possible. We discuss commonly used loss functions for classification and regression.

Mean squared error

Other algorithms like mean absolute error (MAE) and mean bias error (MBE) have been used but they were not so popular as MSE for the task of the regression problem. MSE (Equation 0.17) measures the average squared difference between estimates and actual data. The average magnitude of error is calculated irrespective of their directions. Squaring the term penalizes estimates that are smaller (low) with predictions deviating much more in comparison to real values. This method again is commonly used (Yamashita, Nishio, Do, & Togashi, 2018) because they have a gradient descent

algorithm that is used to minimize loss and provide optimal parameters indeed.

$$mse = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \quad (0.17)$$

where,

n denotes the number of samples,

y_i denotes the prediction value of output,

x_i denotes the correct value of output

Cross entropy loss

Also known as log loss. It measures the performance of a classification model whose output prediction probability value changes between 0 and 1. This is a common loss function method used for multiclass classification (Yamashita et al., 2018). Log loss (Equation 0.18) is another version of the likelihood function with logarithms.

$$L = -(y * \log(p) + (1 - y) * \log(1 - p)) \quad (0.18)$$

where,

y denotes the prediction value of output,

p denotes the probability of assigning an instance to the correct output.

It penalizes heavily the predictions that are confident but wrong. Softmax with one hot encoder applies the logistic classification concept for the loss to predict multi classes classification target labels. Given an appropriate learning rate, a stochastic gradient descent optimizer minimizes total losses during training by running a sufficient number of iterations known as epochs.

3.3.3 Convolutional Neural Networks

This is another kind of neural network which is more applied for image classification when compared to ANN feedforward network (MLP). CNN has the flexibility of providing self-learned features. ANN has difficulties in finding proper weights and activation functions if the number of features in the data is large. This huge input feature may lead to overfitting. Therefore CNNs apply feature dimensionality reduction algorithms to ex-

tract features by adding more convolution layers and pooling layers which will reduce the number of parameters (learned weights). ANN classifier is then applied to make predictions over a large amount of data (e.g visual object and image) (Cormack, 1971; Abiodun et al., 2019). ANN model overfits for high dimensional features (e.g image). CNN is a deep learning technique to take an image as input, assign importance (learnable weights and biases) to various objects in the image, and can distinguish these objects from each other. Studies by (Dufourq et al., 2021; jian Xie et al., 2018; Sankupellay & Konovalov, 2018; Abiodun et al., 2019; Incze, Jancsó, Szilágyi, Farkas, & Sulyok, 2018); have all shown the success of CNN's models in the classification prediction with a good performance by applying data augmentation.

Before using the CNNs model; the task of labeling data is required and it is time-consuming. On the other hand data collection is costly. So time taken in labeling data and cost associated with data collection are the two main challenges. CNNs are made with three types of layers or main building blocks of convolution layers, pooling layers, and fully connected layers (Yamashita et al., 2018). Features are extracted with convolution and pooling layers. The fully connected layer does the classification prediction by mapping the input feature extracted to the desired output.

3.3.4 Convolutional layers

This is the first component of CNN responsible of feature extraction (Rawat & Wang, 2017) using mathematical operations. This process is known as discrete convolution and it produces feature maps by applying different kernel sizes (2x2, 3x3, 4x4, etc). The kernel must be $n \times n$ type where n is a positive integer more than one. Kernel (Figure 0.13) divides inputs images into small slices (receptive fields) to help the feature extraction task (Khan, Sohail, Zahoor, & Qureshi, 2020).

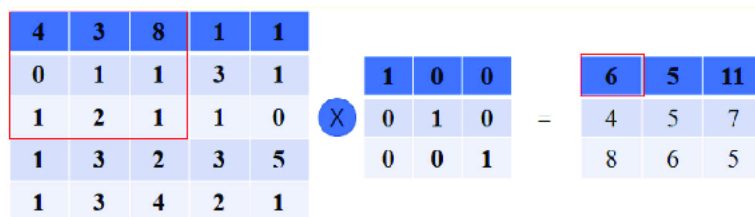


Figure 0.13: Kernel example with stride of 1 (Input, kernel and output).

Figure in 0.13 shows the kernel filter ³.

Kernels (filters) contain a small array of weights (2D) of numbers that are mixed with inputs images matrix to produce feature maps (multiplication of kernel and input elements and sum up them). When using a small kernel, the computation becomes simple without further decomposition operation and the flow of data is simplified. The stride defines the distance between two successive kernel positions and it is usually 1. Feature map loses a small amount of information compared to the original input therefore padding operation is applied to avoid loss of such information. Padding increases the height and width of the input image such that the output stays the same as the original image. Feature maps also depend on the size of the kernel applied. These outputs (feature maps) need to pass to nonlinear functions like ReLU. A repeated convolution and pooling layers may be applied several times to compute different feature map (Rawat & Wang, 2017).

3.3.5 Max pooling

Pooling provides down-sampling and reduces the dimensionality of the feature maps. It is therefore translation invariance (LeCun et al., 2015). It is not affected by a small change in shift, distortions, and a decrease in the number of subsequent learnable parameters which are the weights that are learned during the training phase. Max pooling layer is the type of layer to do the pooling operation. It selects the biggest value in the reconstruction of an image. For instance, at the max-pooling layer of below figure 0.14, each filter which is the box with its own color will take the maximum number in each box. Then the maximum value is added into a new output box with a size of 2x2 pixels.

³Kernel example https://www.researchgate.net/figure/A-toy-example-of-convolution-operation-in-CNN-with-stride-size-as-1-in-which-the-left_fig1_333180752

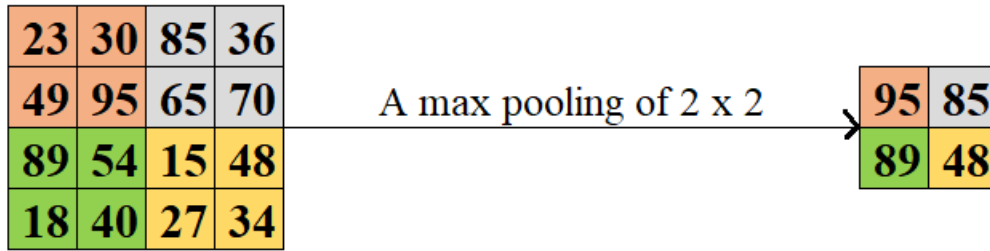


Figure 0.14: Example of max pooling layer

The left part of figure 0.14 represents a single input image matrix of the shape (4,4). It is therefore downsampled by applying a max-pooling layer to produce an output image (figure 0.14 on the right) of the shape of (2,2). The corresponding colors indicate the input region together with the output from that region. Extracted patches from the input feature maps are considered by pooling operation and it chooses the maximum value in each patch while discarding the rest.

3.3.6 Fully connected layer

The feature map from the feature extractor is flattened and sent to a fully connected layer. Extracted features maps at the output at this stage carrying out the task of global operation of classification and here nonlinear combination/operations of inputs features are applied. The activation function changes depending on the target variable (regression or classification).

3.3.7 Evaluating model performance

CNN model for classification is trained, tested, and evaluated. Evaluation of model involves a number of performance metrics that some are explained in the next paragraphs of this section.

Measurement of performance is made due to the fact that there is no guaranty of learning models to have exact predictions with actual values, difficulties in comparing analytical methods between the two values (predicted versus actual). The methods are quite different even though the learned methods perform better on many samples (Jiao & Du, 2016). Algorithm performance measure uses statistical measures to show the progress results from used predictors or learning models (probability of exact predic-

tion) (Jiao & Du, 2016). The data is partitioned into three separate and independent portions (training set, validation set, and testing set).

Training sets are used to create/fit the model and normally it takes 75% of training data. The validation set takes the remaining 25% of training data to evaluate the model performance. Another example of the splitting of data is to take 80% for training, 10% for validation, and 10% for testing. The testing data should come from unseen (new) testing data to test the final model for deployment and use. Samples are chosen randomly to avoid an overfitting scenario. The validation set is used to find optimal parameters of the best model of the classification.

The test set is taken in such that there is a high variation on estimated performance at several times of testing the learned model. The fact that the weights in a neural network are randomly initialized when training from scratch. Thus multiple runs allow one to average the results. The multiple times of runs provide better performance on average. Running the test more than 30 times provides normal and reliable results (Krithikadatta, 2014).

Researchers on the classification prediction model must report on the performance measure of used classifiers algorithms. The need is to understand the mechanism and conditions used to measure the performance in the first place. The classifier may be a “binary classifier” where it assigns one label of the two available classes. “Multi-class classifiers” have more than two labels in their target variable. Binary classification predicts one of the two classes of the target variable. This concept from the binary classification is borrowed and used to model multi-label classification by considering one label versus all remaining labels. Algorithms metrics are measured on basis of counts of observed data (actual and predicted). A confusion matrix is an $n \times n$ contingency table and n represents the number of classes available for the output variable. Below is an example of 2x2 contingency table 0.1:

		Predicted		
		P	N	
Actual	P	TP	FN	sensitivity
	N	FP	TN	specificity
		Precision	NPV	Accuracy

Table 0.1: Confusion matrix

Some researchers may show the performance metrics of the prediction model using errors or a visual representation graph of the area under the curve (AUC or ROC). Only the accuracy, recall, precision, and f-measure are the performance measurements used to evaluate the model classification of this experiment. Accuracy is a common metric measure of classification problems but it is not the only best measurement to evaluate the model with an especially imbalanced dataset. The majority classes have more power or influence on accuracy compared to minority classes. Therefore additional performance metrics such as precision, recall, and f-measure (f1 score) are alternatively used to support accuracy obtained.

- Precision counts the number of positive class predictions that belong to the actual positive class.
- Recall counts the number of positive class predictions done out of all positive samples in the dataset.
- F-Measure is a single score that balances both the concerns of recall and precision in one performance metric number.

Four methods will provide enough and necessary information to evaluate CNNs classifier with and without data augmentation techniques.

- TP (True Positive) defines the number of correctly labeled positive samples.
- FP (False Negative) defines the number of negative samples incorrectly labeled as positive.
- TN (True Negative) defines the number of correctly labeled negative samples.

- FN (False negative) defines the number of positive samples incorrectly labeled as negative.

Function to compute respective metrics for accuracy (Equation 0.19), precision (Equation 0.20), recall (Equation 0.21) and f1-score (Equation 0.22).

$$Accuracy = (TP + FN)/(TP + FP + FN + TN) \quad (0.19)$$

$$Precision = TP/(TP + FP) \quad (0.20)$$

$$Recall = TP/(TP + FN) \quad (0.21)$$

$$F1 = 2 * Precision * Recall / (Precision + Recall) \quad (0.22)$$

3.4 Data augmentation

A number of approaches for data augmentation were discussed in the literature review in chapter 2. This study focuses on the 4 data augmentations method and we also present a baseline method so that the 4 augmentation techniques can be compared to this. These 4 augmentation techniques are based on masking and noise addition.

The baseline performs the duplication of samples to make new copies of samples, masking technique puts the lines (horizontal, vertical) on the original spectrograms, and the noise (random, Gaussian) addition makes a mixture of two images spectrograms (noise, and segment of the pin-tailed or cape robin-chat call). The next sections and subsections provide the description of the baseline method, the two masking augmentation techniques, and the two noise addition augmentation techniques. The description will also make the difference between masking techniques (i.e frequency and time), and noise addition techniques (i.e random noise and Gaussian noise). The purpose of the augmentation techniques is to increase the number of spectrograms from an initial number of examples to a larger number of examples.

The collected and pre-processed dataset is imbalanced. The training set (82 audio files) has 952 samples of Cape robin-chat calls, 1881 samples of pin-tailed Whydah calls and 4557 samples of noises. The augmentation techniques are applied to address this class imbalance. The baseline method is simply a duplication of randomly selected

examples. In this thesis, after applying the baseline method the resulting augmented dataset contains 2,000 examples. These 2000 examples are from the 500 examples of Cape robin-chat and 1500 newly created copies of Cape robin-chat spectrograms with augmentation, or 500 of pin-tailed Whydah and 1500 newly created copies of these pin-tailed Whydah spectrograms with augmentation, or 500 of noises and 1500 newly created copies of these noises spectrograms with augmentation. The noise addition augmentation techniques generates new copies of audio segments by randomly adding Gaussian noise to a given segment of audio. Thus, segments of audio from any of the three classes can be augmented. Time masking selects a random range of time periods to be masked from the original image. Frequency masking hides a random range of frequencies from the original image.

A free account on Google Colaboratory was used to train the neural network. This type of account is limited in terms of the resources (GPU run time available per day). The proposed experiment was conducted as follows. For each augmentation technique a random number of samples from each class was selected (1881 of the pin-tailed whydah, 952 of the cape robin-chat and 4557 of the noise class). Then, the segments from the pin-tailed whydah and noise class were downsampled to 952 classes to create a balanced dataset. The augmentation techniques were then applied to this balanced dataset. Let say that we want 2000 samples in each category after one augmentation technique, therefore, we can draw 500 samples in each category and augment those 500 samples to 2,000 samples. This will result in 500 original samples and 1500 augmented samples. The total samples will be 6000 after augmentation in all categories (2000 samples of pin-tailed whydah call, 2000 samples Cape robin-chat call, and 2000 samples of noise).

All these 6000 samples are used to train and validate the model performance on that particular augmentation technique selected. The training set with 75% of 6000 samples will be 4500 samples in size and 25% remaining of samples are taken for validation data (i.e 1500 remaining samples). The testing set is from the testing audio files of unseen data. The size of samples taken for the testing set is 4500 samples from testing audio files. The testing set is not augmented.

A total of 2000 samples of each category was chosen because the amount of RAM provided on the free account on Google Colaboratory could not process samples beyond 2000 samples for each category. The 500 samples are referred to as the initial/original samples. The study encourages us to select a small number of initial samples. A 500

samples was taken in the study as one example but the study has experimented the 250 samples and the 100 samples. Based on the previous explanation, the experiment uses the same original samples and they are augmented using different techniques. Thus, in each case, the experiment starts from the same original samples, however, the experiment applies different augmentation techniques as a means of comparing the techniques. This is done to avoid adding new random samples into the original samples to allow for fair comparisons to be made and we have a better comparison of results from different augmentation techniques. This can be repeated for any starting value and not necessarily 500.

3.4.1 Baseline method

The evaluation of the model will be with and without the data augmentation techniques. The baseline method produces results for the model evaluation without augmentation. RAM size limitation forces us to use a total of 2000 samples in each group.

It starts with a fixed number of samples for all species. The baseline method creates copies without the modification on the initial samples but the 4 remaining techniques of data augmentation do the changes over original data. For example, we draw 500 samples from each group (952 pin-tailed whydah, 1881 Cape robin-chat, 4557 noises); in the baseline method, the drawn 500 samples are duplicated 4 times to make 2000 samples in each group. If we draw 250 samples the duplication will be 8 times and 20 times for 100 drawn samples. Each approach is discussed in detail in the next paragraphs.

3.4.2 Noise addition

Adding random noises

Having calls of 1881 pin-tailed whydah, 952 Cape robin-chart, and 4557 noises; we draw in each group let say 500 initial samples. The injection of noise technique will be selecting a segment of audio from the noise call (one of the 500 samples of noises is chosen randomly) and then adding it to a segment of audio that contains a call(one of those 500 initial samples of pin-tailed whydah or Cape robin-chat). The 2000 samples are total samples of each group after the augmentation process. The 500 initial samples and 1500 were newly created by the random noise augmentation technique. Mixing up the noise

and initial data is done by summing up the two vectors which contain amplitude values from both and making an addition of them to create new copies of samples.

Figure 0.15 illustrates a spectrogram that contains calls from the pin-tailed whydah. We will illustrate the various augmentation techniques after they have been applied to this spectrogram. Figure 0.16 illustrates the result when random noise was added to the spectrogram from figure 0.15.

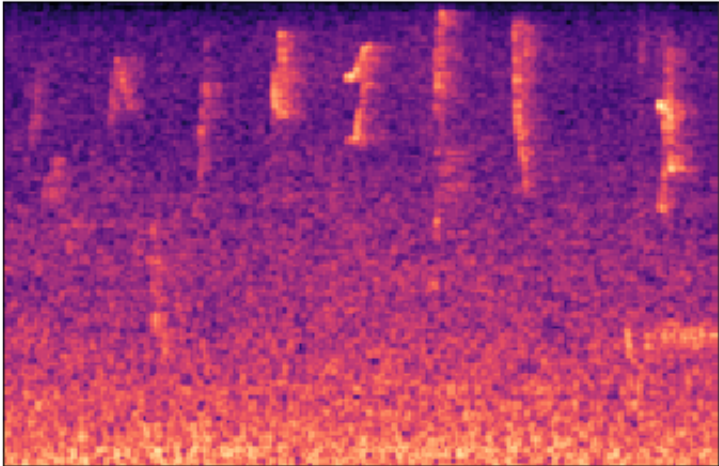


Figure 0.15: original image of pin-tailed whydah before augmentation

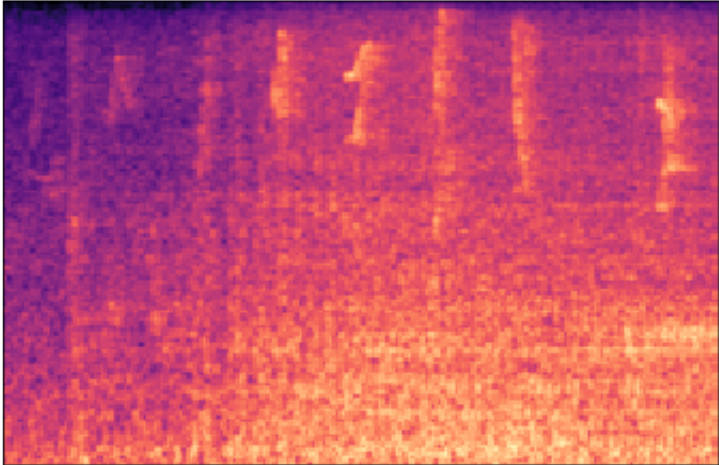


Figure 0.16: Random noise addition applied to pin-tailed whydah

Gaussian noise addition

In this technique, we create a random normal vector which contain amplitude values with three parameters; mean of 0.0 (loc=0.0), the standard deviation of 1.0 (scale=1.0), and size which is the shape of the original signal. The signal created is called noise and it may be multiplied with a scalar to control the level of associated noise. The scalar we used is 0.17 in our case for not producing a new copy with a lot of noise. One could not see the call if there are too much noise created. Having noise and original signals vector we add them to create a noisy signal. Figure 0.15 illustrates a spectrogram of the call from the pin-tailed whydah before augmentation, and the figure 0.17 is an illustration spectrogram after applying Gaussian noise addition augmentation.

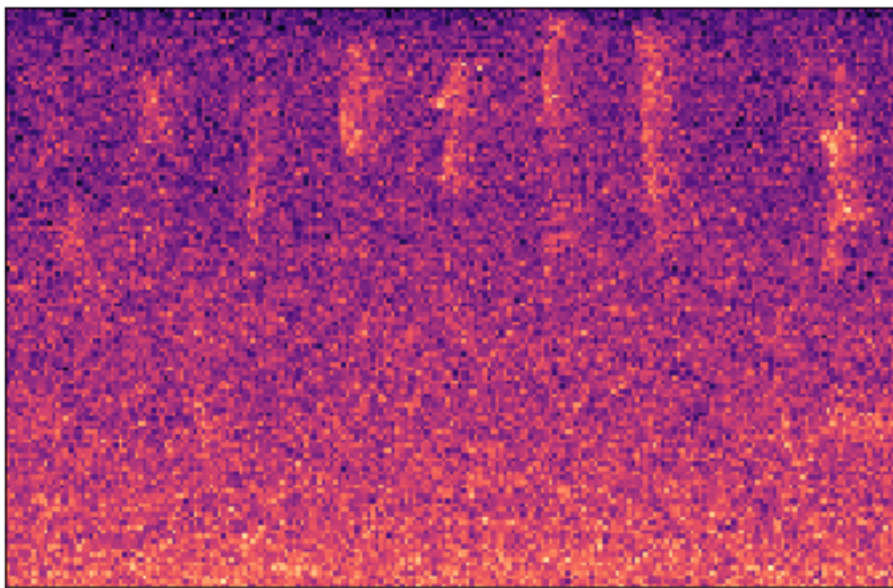


Figure 0.17: Gaussian noise addition applied to pin-tailed whydah

The difference between random and Gaussian noise addition

Gaussian noise and random noise addition are both noise addition techniques. The difference is that the random noise addition picks a random noise sample from 4557 noises samples and add it to the call from either 1881 pin-tailed whydah or Cape robin-chat, while the Gaussian noise addition will create a new random noise from the normal distribution and it is added to the call from either 1881 pin-tailed whydah or Cape robin-chat.

Spectrogram images are a 2D matrix of pixels values. The Gaussian noise creates a random array of values which are normally distributed with the mean of 0.0 and variance of 1.0. This array of values is Gaussian because of the use of the mean of 0.0 and the variance of 1.0 to create the specified array of values. This array of values is transformed to 2D matrix values of similar shape (format) to the segment of the call of pin-tailed or Cape robin-chat. The use of the same shape/format of the image helps in summing up the noise matrix values and the matrix of the segment of pin-tailed or Cape robin-chat call. The sum of two matrices may result in an image spectrogram with high noise compared to the segment of the call. These Gaussian generated noises are further controlled by multiplying the sum of two matrices by a scalar to reduce the noise from the call of pin-tailed whydah or Cape robin-chat. The scalar of 0.17 was used to serve this purpose.

3.4.3 Masking

Masking is the process of hiding a small region of a spectrogram image. Frequency and time are the two known maskings used in this research.

Time masking

This augmentation technique generates new examples by randomly placing one vertical line on the spectrogram. A random range of a small period of time on the spectrogram image of the original image is selected to be hidden (Park et al., 2019). Time masking uses one vertical line in the experiment because of the size of the spectrogram image and the maximum call duration. The size of the input spectrogram image to CNN is fixed to the height of 126 and the width of 214. The maximum call duration is 3 seconds. With this maximum call duration and fixed size of the spectrogram, the masking must be controlled in such a way the augmented image will not lose much information. Masking must not make the call invisible, and it cannot go beyond the spectrogram image. The experiment proposes to put one line to avoid producing a noisy spectrogram which may produce poor performance. So the experiment suggests one line as seen in figure(Figure 0.18) to have good performance in the prediction classification. See the (Figure 0.18) below for more details:

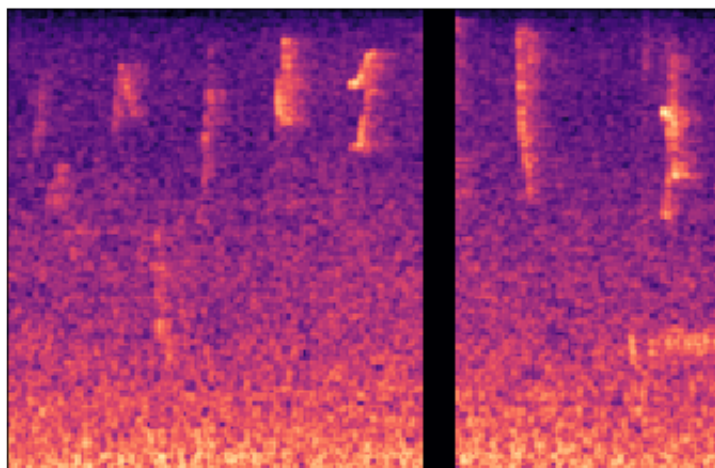


Figure 0.18: Time masking applied to pin-tailed whydah

Frequency Masking

This augmentation technique generates new examples by randomly placing one horizontal line on the spectrogram. A random range of frequencies on the original spectrogram image is chosen to be hidden (Park et al., 2019). The next image (Figure 0.19) shows details:

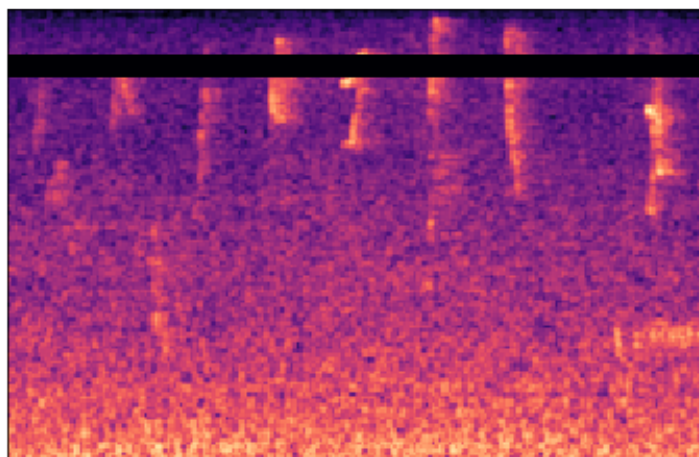


Figure 0.19: Frequency masking applied to pin-tailed whydah

Chapter 4 Results and discussion

4.1 Introduction

This chapter presents and discusses the results of the study regarding the aim of the research, which is to explore data augmentation techniques for small bioacoustics datasets and control the effect of applied methods on different sizes of samples in classification prediction performance metrics.

Duplication of samples, masking (time and frequency), and injection of noise (Gaussian, random) were the main four methods of data augmentation used for this research to balance and increase the dataset. Each of these techniques has been tested with 100, 250, and 500 original samples. Augmentation is made up of 2000 samples from each group of original samples to see the behavior of the applied augmentation technique in the model classification performance. Results from 4 augmentation techniques and the baseline for each group of initial samples (100, 250, 500) are compared. For example, if the experiment takes 250 samples i.e it is taking 250 original samples of pin-tailed whydah augmented to 2000 samples. Same to Cape robin-chat and noises. Performance metrics are reported in a summary table with only prediction accuracy and, f1-score since it is made from precision and recall. There are two main sections in this chapter. The first one discusses the model training and hyper-parameter tuning and, the second part is about the results of the findings and their comparison.

4.2 Model training and hyper-parameter tuning

The audio files were pre-processed on a local computer while the remainder of the steps were carried out on Google Colab which provides a GPU. The rest of the work such as CNN model training and classification has been done using the Google Colab plat-

form with Colab free account. Colab runs TensorFlow operations using (GPU) which is speeding up the computation process. It also provides 12 GB of RAM. The limited size of RAM has changed the plan initially made on the data to be executed in building the CNN model. The 2000 samples for each species mentioned above were due to limitations in RAM, only 6000 samples in total could be loaded and not more than that.

Taking 82 audio recordings for training and again 82 audio for testing was a preliminary requirement to run the model. The selection of these audio files was done by using a chronological order so that no same day was used for training and testing sets. These 82 recordings on training gave 952 Cape robin-chat, 1881 pin-tailed whydah, and 4557 noises. In each of these numbers, the experiment picks randomly initial pre-defined samples to be used for augmentation of up to 2000 samples. Training results with 6000 samples and the data are split into 4500 samples for CNN model training and 1500 samples for the validation data. In the same way, the experiment took again 4500 samples for the testing data from the testing audio files of unseen data.

The experiment conducted a hyper-parameter search to find suitable hyper-parameters of the training model. These parameters were selected by conducting a random search over a number of different values. The hyper-parameter optimizer used was Adam.

Adam (Adaptive Moment Estimation) optimizer is an adaptive variant of SGD (Stochastic Gradient Descent) that outperform well for the certain complex task of deep learning compared to SDG (J. Zhang et al., 2020). An SDG is a method used to train neuron networks to find optimal parameters of the model. These parameters are updated iteratively during the model training by comparing each set of parameters with the associated loss. At the end of the predefined iteration parameters with minimum loss will be considered optimal parameters of the model. Adam is used since it has shown good performance in other research on neural networks.

The learning rate parameter was 0.01, and the number of epochs was 20. This has been done by the addition of layers and calibrating the convolution layers needed and max-pooling layers. Table 0.1 indicates a summary of the configuration of CNN. Table 0.1 shows the training model with the number of network parameters at each layer.

No	Layer	Activation Function	Filter		Network Parameters
			Number	Size/ Shape	
0	Conv2D	ReLU	32	(3, 3)	320
1	Conv2D	ReLU	32	(3, 3)	9248
2	Conv2D	ReLU	32	(3, 3)	9248
3	Conv2D	ReLU	32	(3, 3)	9248
4	Conv2D	ReLU	32	(3, 3)	9248
5	Flatten	ReLU	-	-	0
6	FC (Dense)	ReLU	32	-	8224
7	FC (Dense)	Softmax	3	-	99

Table 0.1: Training CNN model parameters.

A conv2D in table 0.1 stands for a 2D convolution layer and the filter of this conv2D has a height and width. FC in table 0.1 stands for the fully connected layer and it connects all previous output nodes to the next layer nodes. The training model with the configuration shown in the table 0.1 takes around 665 seconds (11 minutes) for the 6000 dataset samples. This is the minimum time of the experiment in the frequency masking of 250 initial samples. The maximum time used from all experiments is 1379 seconds (22 minutes). This time was given by the execution of the Gaussian noise addition with 250 initial samples. The minimum time from all experiments was 664 seconds (11 minutes). This period of 11 minutes was given by the frequency masking with 250 initial samples. Other experiments were executed within the maximum and minimum time. All experiments have given 1031 seconds (17 minutes) on average. So the training time

can be estimated to 1031 seconds (17 minutes). The experiment may take around 17 minutes to train the model. Model training together with prepared data has produced results of findings that are shown in the next section of this chapter.

4.3 Results of the findings and their comparison

Based on the results from table 0.2. The table shows a summary of results starting with a sample size of 100; 250 and 500 original data each augmented to 2000 samples on each species (CRC, PTW, and NOISE). PTW stands for pin-tailed whydah call, CRC is Cape robin-chat call and NOISE represents any call other than these two calls. Associated performance metrics measured using f1-score and accuracy are recorded in the table. The f1 score is shown because it involves recall and precision. The nan values of table 0.2, are due to the division by zero in the f1 score formula at the denominator.

			Baseline	Frequency Masking	Time Masking	Gaussian Noise	Random Noise
100 samples	Accuracy	Train	1.000	1.000	0.999	0.985	0.333
		Test	0.858	0.832	0.839	0.844	0.222
	F1 Score	CRC	0.784	0.775	0.769	0.785	nan
		NOISE	0.883	0.858	0.866	0.875	nan
		PTW	0.842	0.807	0.818	0.798	0.363
250 samples	Accuracy	Train	1.000	0.999	0.997	0.886	0.508
		Test	0.873	0.882	0.874	0.851	0.584
	F1 Score	CRC	0.849	0.859	0.832	0.800	0.623
		NOISE	0.920	0.902	0.896	0.869	0.586
		PTW	0.861	0.847	0.850	0.789	0.568
500 samples	Accuracy	Train	0.995	0.991	0.977	0.961	0.857
		Test	0.899	0.895	0.903	0.878	0.895
	F1 Score	CRC	0.872	0.863	0.847	0.868	0.846
		NOISE	0.919	0.915	0.922	0.902	0.917
		PTW	0.867	0.864	0.884	0.826	0.868

Table 0.2: Comparison between experiment techniques versus sample size.

The comparison table 0.2 shows that the baseline and time masking, are the best techniques using f1-score. The values in bold are the best predictions in each initial sample size (100, 250, and 500) of the three call types for the used experiment techniques. For example, a 100 initial sample size has an f1-score of 84.2% as the best prediction of pin-tailed whydah for the used techniques. This prediction is achieved by the baseline technique. Both methods (baseline and time masking) work well on the small initial sample size (100) and the big initial sample size as well. With the constraints explained in the model training and hyper-parameter tuning subsection; the 2000 samples were a fixed number of samples to make an experiment of the study. 100, 250, and 500 samples were taken as original samples. Augmentation is improving model performance from the use of 100, 250, and 500 original samples. The use of 100 original samples is a good approach to explain the aim of the study. A small dataset of 100 original samples augmented to 2000 samples and it can produce a 78.5% of f1-score in classifying Cape

robin-chat with Gaussian noise addition and, an f1-score of 84.2% with the baseline to classify pin-tailed whydah; this shows that the augmentation has a statistical significance in improving model performance without re-collecting the data which may involve additional unplanned cost. The model has improved its performance by applying data augmentations on 100 which is a small dataset compared to 250 and 500 initial samples to start with. A 100 sample is 20 times lower compared to 2000 samples. A 100 taken samples are a small dataset and it satisfies the aim of the research with the success on the two species.

The random noise addition augmentation technique is the worst method for the small amount of data (sample less than 500 initial samples). It has a very poor performance and cannot be used for classification prediction. The study aim is to work with a small sample size which also does not require a lot of processing resources (e.g RAM). Small initial sample size has an advantage since it runs with a free Colab account.

The testing accuracy keeps on increasing as the initial sample size is increasing on various techniques. This proves the hypothesis of the study. It states that an increase in the sample size increases accuracy. The best testing accuracy in the next initial sample size is greater than the best accuracy of the previously taken initial sample size. For example, a 100 sample's best accuracy is 85.8% which is less than the one of the 250 samples (88.2%).

For the baseline method, frequency and time masking, and Gaussian noise augmentation techniques may be used alternatively in the experiment. All have a slight change in the prediction accuracy. The change is not significant for a small initial sample size (100). The fact of producing good results at the small sample size makes them be the good techniques. It is, therefore, possible to monitor these two species (pin-tailed whydah and Cape robin-chat) on a small scale of data captured. Any of the four techniques mentioned in this paragraph can be used to help the work of ecologists in monitoring both species. The overfitting is explained with the gap spaces between the curves. Accuracy and loss curves are given for training and validation sets.

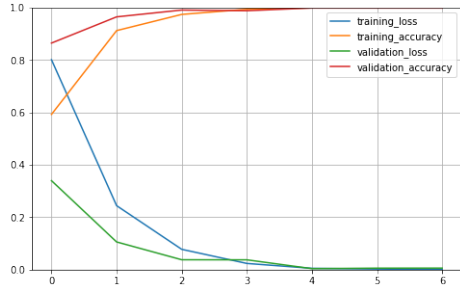


Figure 0.1: Frequency masking evaluation plot for 100 sample.

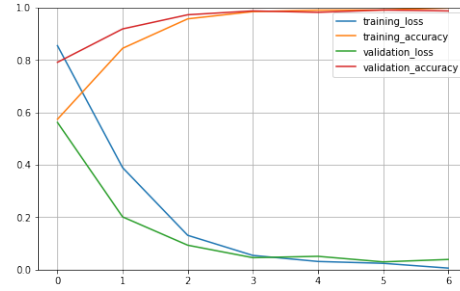


Figure 0.2: Time masking evaluation plot for 100 sample.

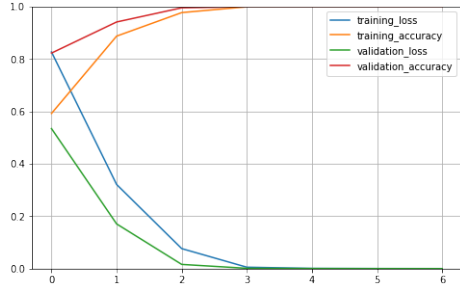


Figure 0.3: Baseline evaluation plot for 100 sample.

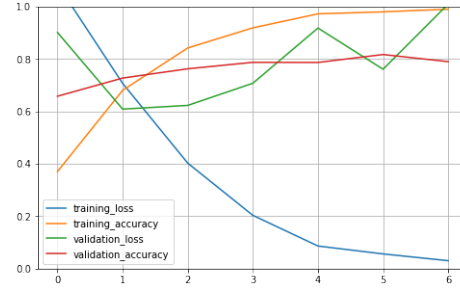


Figure 0.4: Gaussian noise addition evaluation plot for 100 sample.

The first three figures (0.1, 0.2, 0.3) indicate that each of the three techniques is not producing overfitting, meaning the curve of the validation loss is less than training loss, and each of the three techniques has a validation accuracy more than the training accuracy. The three first techniques outperform well. Baseline, time, and frequency masking start with a small difference in their results and they stabilize at around epoch 2 (after two epochs on the x-axis). The Gaussian noise addition can be used at around 1 epoch (on the x-axis) of the fourth figure 0.4. This figure of 0.4 shows that the Gaussian noise addition has most of the time overfitting. It changes more with validation loss greater than training loss thus creating less accuracy for validation data. This situation starts from the second epoch on the x-axis. The Gaussian can still be valid since the history checkpoints will choose the best parameters where the technique will produce a model without overfitting. Weights selected at around 1 on the x-axis of Gaussian noise addition will be kept for the best model. For 500 initial samples, the three first techniques continue to stabilize results without overfitting. The Gaussian noise addition remains with overfitting but is being reduced (see figure 0.6). The random noise addition that performed poorly in the 100 initial samples shows that the 500 original samples had produced good results without overfitting (see figure 0.5).

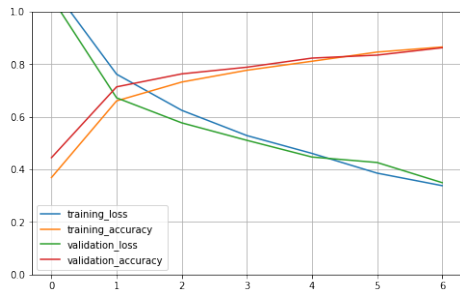


Figure 0.5: Random noise evaluation plot for 500 sample.

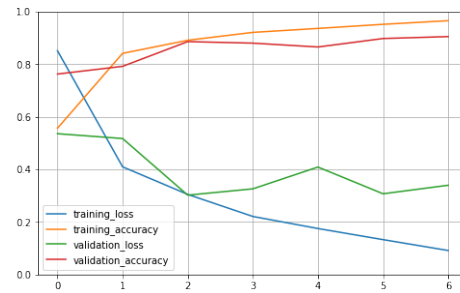


Figure 0.6: Gaussian noise evaluation plot for 500 sample.

At least 500 initial samples are required for the classification prediction of 80% and above for the three types of call if one needs to use any of the 5 techniques of augmentation (baseline, frequency or time masking, or noise addition) used in this experiment. The testing accuracy increases as the size of initial sample size are increasing for all augmentation techniques.

Chapter 5 Conclusions and recommendations

5.1 Conclusions

It is an accepted fact within the machine learning community that a large amount of data is required to train CNNs and that training CNNs from randomly initialized weights on small datasets is not recommended. The experiment results suggest that on those two species (pin-tailed whydah and Cape robin-chat), good performance can be achieved.

Analysis of survey data obtained in this research shows that an increase of samples with data augmentation improves the performance accuracy. The study aimed to test if a small samples size applied with data augmentation performs a better accuracy. Applied methods with different samples size (100, 250, and 500 samples) have proven this hypothesis. Baseline, frequency and time masking, and Gaussian noise injection are augmentation techniques to prove the stated hypothesis. There is a significant change when increasing sample size.

The random noise injection augmentation technique becomes unusable. On less amount of sample size, it performs poorly while on a huge amount of samples other methods work well in the classification prediction. It will require huge computation resources (RAM, GPU, and storage) since it may produce good results on high samples. Other methods work well without additional resources which may require an additional cost of payment.

The baseline method fits well the data on the small size of samples. Masking (time and masking), and Gaussian noise addition may be applied since there is no big difference in results. They bring small change which is not significant enough for potential

improvements. Results are very close to each other. Time masking is the best model with the highest accuracy of the test. It may work on small and big datasets.

The CNN takes a couple of few minutes to train the model when compared to the time taken by humans to train on the three types of calls. A human requires an expert for the training and it takes many hours to train a human annotator on recognizing the calls. On CNN, humans take time to prepare inputs data but the training is short in time. The results of the experiment show the success of the classification of the pin-tailed whydah and Cape robin-chat for passive acoustic monitoring. Ecologists are strongly recommended methods used by the experiment to monitor other species too.

5.2 Recommendations

Based on the experiment of the study, the experiment suggest baseline and time masking methods for the classification prediction since they have performed better results for small and big size data.

The study recommends also a combination of these data augmentations and compare results. This could not be done on Colab free account which provides limited resources to run the model. Additional costs may be applied to get full resources.

The best approaches of these techniques work well on fewer samples, therefore, the baseline, masking (time and frequency), and Gaussian noise addition are the best models. They achieve good results on small sample size (see 100 samples). It is an advantage for working with a small dataset on a free account of Colab but it is a barrier if 2000 samples are changed to another number of samples beyond. Free account of Colab is limited in handling these samples. While the results were encouraging, further investigations are required in comparing the augmentation techniques to different sample sizes. This would, however, require that additional computational resources be available. The study is limited to two species (pin-tailed whydah and Cape robin-chat) but the collected data contain a lot of calls that can be used by further researchers since the data has been made publicly available. This repository is recommended to the researchers with the study that may require these data without being recollected. The experiment of the study has shown success but further research in this field of bioacoustics may expand these machine learning/deep learning techniques to other species. The results presented in this thesis and the findings from the literature reveal that machine learning can successfully be applied to passive acoustic monitoring. The encouraging results indicate that these

techniques could be implemented in practice to monitor various species, for example, it could be implemented to monitor and conduct biodiversity assessments in various locations, including Rwanda.

References

- Abiodun, O. I., Kiru, M. U., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., ... Gana, U. (2019). Comprehensive review of artificial neural network applications to pattern recognition. *IEEE Access*, 7, 158820-158846.
- Bermant, P. C., Bronstein, M. M., Wood, R. J., Gero, S., & Gruber, D. F. (2019). Deep machine learning techniques for the detection and classification of sperm whale bioacoustics. *Scientific reports*, 9(1), 1–10.
- 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.
- Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 1429–1441.
- Browning, E., Gibb, R., Glover-Kapfer, P., & Jones, K. E. (2017). Passive acoustic monitoring in ecology and conservation.
- Çakir, E., Adavanne, S., Parascandolo, G., Drossos, K., & Virtanen, T. (2017). Convolutional recurrent neural networks for bird audio detection. *CoRR*, abs/1703.02317. Retrieved from <http://arxiv.org/abs/1703.02317>
- Cape Robin-Chat - eBird*. (n.d.). <https://ebird.org/species/carcha1>. (Online accessed on: 2021-08-26)
- Cormack, R. M. (1971). A review of classification. *Journal of the Royal Statistical Society: Series A (General)*, 134(3), 321–353.
- Devries, T., & Taylor, G. W. (2017). Dataset augmentation in feature space. *ArXiv*, abs/1702.05538.
- Dufourq, E., Durbach, I., Hansford, J. P., Hoepfner, A., Ma, H., Bryant, J. V., ... others (2021). Automated detection of hainan gibbon calls for passive acoustic monitoring. *Remote Sensing in Ecology and Conservation*.
- Feng, J., & Lu, S. (2019). Performance analysis of various activation functions in

- artificial neural networks. In *Journal of physics: Conference series* (Vol. 1237, p. 022030).
- Geng, M., Xu, K., Ding, B., Wang, H., & Zhang, L. (2018). Learning data augmentation policies using augmented random search. *ArXiv, abs/1811.04768*.
- 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. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2468067219300306> doi: <https://doi.org/10.1016/j.ohx.2019.e00073>
- Incze, A., Jancsó, H.-B., Szilágyi, Z., Farkas, A., & Sulyok, C. (2018). Bird sound recognition using a convolutional neural network. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)* (pp. 000295–000300).
- Jackson, P. T. G., Atapour-Abarghouei, A., Bonner, S., Breckon, T., & Obara, B. (2019). Style augmentation: Data augmentation via style randomization. In *Cvpr workshops*.
- Jian Xie, J., qing Ding, C., Li, W., & Cai, C.-H. (2018). Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks. *ArXiv, abs/1803.01107*.
- Jiao, Y., & Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*, 4(4), 320–330.
- Jiao, Y., Tu, M., Berisha, V., & Liss, J. (2018). Simulating dysarthric speech for training data augmentation in clinical speech applications. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6009–6013).
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 1 - 62.
- Koh, C.-Y., Chang, J.-Y., Tai, C.-L., Huang, D.-Y., Hsieh, H.-H., & Liu, Y.-W. (2019). Bird sound classification using convolutional neural networks. In *Clef (working notes)*.
- Koskimies, P. (1989). Birds as a tool in environmental monitoring. In *Annales zoologici fennici* (pp. 153–166).
- Krithikadatta, J. (2014). Normal distribution. *Journal of conservative dentistry: JCD*, 17(1), 96.
- Lasseck, M. (2018). Audio-based bird species identification with deep convolutional

- neural networks. In *Clef*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.
- McCloughlin, M., Stewart, R., & McElligott, A. (2019). Automated bioacoustics: methods in ecology and conservation and their potential for animal welfare monitoring. *Journal of The Royal Society Interface*, *16*, 1–12. Retrieved from <http://dx.doi.org/10.1098/rsif.2019.0225> doi: 10.1098/rsif.2019.0225
- Mesaros, A., Heittola, T., Benetos, E., Foster, P., Lagrange, M., Virtanen, T., & Plumb-ley, M. D. (2017). Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(2), 379–393.
- Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117-122.
- Nanni, L., Maguolo, G., & Paci, M. (2020). Data augmentation approaches for improving animal audio classification. *Ecological Informatics*, *57*, 101084.
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
- Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019, Sep). Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*. Retrieved from <http://dx.doi.org/10.21437/Interspeech.2019-2680> doi: 10.21437/interspeech.2019-2680
- Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- Pilon, C. D. (2015). *Probabilistic programming and bayesian methods for hackers*. Addison-Wesley Professional.
- Pin-tailed Whydah - eBird*. (n.d.). <https://ebird.org/species/pitwhy>. (Online accessed on: 2021-08-26)
- Potamitis, I. (2014). Automatic classification of a taxon-rich community recorded in the wild. *PloS one*, *9*(5), e96936.
- Priyadarshani, N., Marsland, S., & Castro, I. (2018). Automated birdsong recognition in complex acoustic environments: a review. *Journal of Avian Biology*, *49*(5), jav-01447.
- Ratner, A. J., Ehrenberg, H. R., Hussain, Z., Dunnmon, J. A., & Ré, C. (2017). Learning

- to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, *30*, 3239-3249.
- Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, *29*, 2352-2449.
- Sankupellay, M., & Konovalov, D. (2018). Bird call recognition using deep convolutional neural network, resnet-50. In *Proceedings of acoustics* (Vol. 7).
- Seyfarth, R., & Cheney, D. (2003). Meaning and emotion in animal vocalizations. *Annals of the New York Academy of Sciences*, *1000*.
- Shannon, C. (1949, jan). Communication in the presence of noise. *Proceedings of the IRE*, *37*(1), 10–21. Retrieved from <https://doi.org/10.1109/jrproc.1949.232969> doi: 10.1109/jrproc.1949.232969
- Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017). Research on data augmentation for image classification based on convolution neural networks. *2017 Chinese Automation Congress (CAC)*, 4165-4170.
- Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040-53065. doi: 10.1109/ACCESS.2019.2912200
- Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. In *Bio-inspired neurocomputing* (pp. 203–224). Springer.
- Taye, G. T., Hwang, H.-J., & Lim, K. M. (2020). Author correction: Application of a convolutional neural network for predicting the occurrence of ventricular tachyarrhythmia using heart rate variability features. *Scientific Reports*, *10*(1), 1–2.
- Teixeira, D., Maron, M., & van Rensburg, B. J. (2019). Bioacoustic monitoring of animal vocal behavior for conservation. *Conservation Science and Practice*, *1*(8), e72.
- Tóth, B. P., & Czeba, B. (2016). Convolutional neural networks for large-scale bird song classification in noisy environment. In *Clef (working notes)* (pp. 560–568).
- Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
- Wei, S., Zou, S., Liao, F., et al. (2020). A comparison on data augmentation methods based on deep learning for audio classification. In *Journal of physics: Conference series* (Vol. 1453, p. 012085).
- Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (dicta)* (pp. 1–6).

- Yamashita, R., Nishio, M., Do, R., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, 611 - 629.
- Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S., Kumar, S., & Sra, S. (2020). Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33, 15383–15393.
- Zhang, X., Wang, Z., Liu, D., & Ling, Q. (2019). Dada: Deep adversarial data augmentation for extremely low data regime classification. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2807-2811.
- Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. In *Aaai*.

Appendix A

Plagiarism report



10



Acoustic Data Augmentation for Small Passive Acoustic Monitoring Datasets

By Student Name: NSHIMIYIMANA Aime

¹³ Registration Number: 220003483

A dissertation submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA SCIENCE (DATA MINING)

Activate Windows

Investigating data augmentation techniques for small bioacoustics datasets

ORIGINALITY REPORT

10%
SIMILARITY INDEX

7%
INTERNET SOURCES

5%
PUBLICATIONS

3%
STUDENT PAPERS