# A Study on Machine Learning Techniques for Bird Call Classification

---


Presented by:

Thatohatsi Motlhamme

Prepared for:

Dr Yaaseen Martin

October 24, 2024

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the degree of Bachelor of Science in Engineering in Mechatronics.

# Abstract

This research project investigates the application of machine learning techniques for automated bird species classifications based on their vocalisations. The motivation stems from the need for efficient, non-invasive biodiversity monitoring tools to aid in conservation efforts.

The project includes a comprehensive review of relevant literature that lays the groundwork for the selection of machine learning models and feature extraction methods. The chosen techniques include Random Forest, k-Nearest Neighbours, Support Vector Machines, and deep learning models like Convolutional Neural Networks and transfer learning architectures, specifically VGG-16 and ResNet-50.

The study employed minority class oversampling and various augmentation methods like pitch shifting and noise injection to enhance model robustness. Initial results on imbalanced data show that traditional models, while computationally efficient, struggle to generalise across diverse species, particularly for underrepresented classes. Among traditional models, Random Forest and Support Vector Machines showed moderate performance, achieving accuracies of 72.3% and 74.4%, respectively, on the testing dataset. However, deep learning approaches, particularly transfer learning with VGG-16, demonstrated superior performance, achieving a testing accuracy of 89.2%.

Key findings from the evaluation phase reveal that data balancing significantly improves species representation in training, while augmentation techniques yield limited success in generalising to new audio environments. The VGG-16 model achieved the highest controlled test accuracy at 86.5%. ResNet-50 also performed well, though it showed slightly lower generalisation in unannotated field data, with testing accuracies below 47%.

# Acknowledgments

First and foremost, I would like to thank God for being my ultimate Project Manager throughout this degree, giving me the strength and guidance for completing this project.

I would like to thank my mother for being the foundation upon which all my achievements are built. Her love and support has been my constant compass, and the persistence she has instilled in me has become my greatest tool.

Dr. Yaaseen Martin, my supervisor, your patience and guidance have been invaluable. When I stumbled, your wise words helped me find my footing. Your mentorship has shaped not just this project, but my approach to challenges ahead.

To my course twin, Beenzu, your ability to bring joy to even the most challenging days has been a gift. And to my beloved gang - Nyiko, Kudzi, Mpilo, Nate, and Nanga - what started as classmates has blossomed into a brotherhood I'll cherish forever. We've shared struggles and triumphs, tears and laughter, and through it all, we've lifted each other up. Thank you for the accountability, the endless support, and for proving that the toughest challenges are better faced together. Our friendship has been one of the greatest gifts of this degree.

*"Look at the birds of the air, for they neither sow nor reap nor gather into barns; yet your heavenly Father feeds them. Are you not of more value than they?"*

— Matthew 6:26 (NKJV)

# Plagiarism Declaration

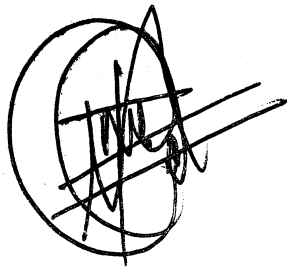
1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.

3. This final year project report is my own work (except where I have attributed it to others).

4. I have not paid a third party to complete my work on my behalf. My use of artificial intelligence software has been limited to **debugging code errors, and grammar checks** (specify precisely how you used AI to assist with this assignment, and then give examples of the prompts you used in your first appendix).

5. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as there own work or part thereof.



Thatohatsi Motlhamme

October 24, 2024

# Acronyms

**ARBIMON** Automated Remote Biodiversity Monitoring Network. 12, 13

**ATP** Acceptance Test Procedures. 64

**AUC** Area Under the Receiver Operating Characteristic Curve. 37, 51, 66

**BER** Band Energy Ratio. 9

**BirdCLEF** Bird Challenge for Listening to Endangered Fauna. 14, 17, 23

**CNN** Convolutional Neural Network. viii, x, 10, 11, 13, 16, 17, 26, 40, 41, 53, 54, 60, 63, 88, 89

**CNNs** Convolutional Neural Networks. 4, 10, 11, 13, 17, 18, 25, 26, 40, 66

**CQT** Constant-Q Transform. 9, 24, 28, 46, 51, 58, 61, 65

**DL** Deep Learning. vi, ix, 4–6, 9–11, 17, 18, 21, 22, 29, 39, 44, 46, 55, 56, 88

**EDA** Exploratory Data Analysis. 32

**FC** Fully Connected. 89

**FFNN** Feedforward Neural Network. viii, ix, 26, 39, 40, 52, 53, 60, 63, 88

**FFNNs** Feedforward Neural Networks. 4, 26

**GMMs** Gaussian Mixture Models. 8

**GRU** Gated Recurrent Unit. 11, 25

**HMM** Hidden Markov Model. 13

**HMMs** Hidden Markov Models. 8

**KNN** k-Nearest Neighbour. viii, 8, 25, 29, 38, 51, 60, 63

**KNNs** k-Nearest Neighbours. 4, 8, 21, 25

**LSTM** Long Short-Term Memory. 11, 25

**MFCC** Mel-Frequency Cepstral Coefficient. 46, 49, 51, 57, 61, 64

**MFCCs** Mel-Frequency Cepstral Coefficients. 8, 9, 24, 25, 28, 34, 58, 61

**ML** Machine Learning. 1–4, 6–8, 11, 13, 17–19, 21, 22, 27, 35, 63–65

**OOP** Object Oriented Programming. 27

**ResNet** Residual Network. 13, 17

**RF** Random Forest. viii, ix, 11, 25, 29, 37, 38, 47–49, 59, 61, 63, 64, 86

**RFs** Random Forests. 4, 7, 8, 25

**RMS** Root Mean Square. 9, 24, 65

**RNN** Recurrent Neural Network. 11

**RNNs** Recurrent Neural Networks. 10, 11, 25

**SVM** Support Vector Machine. viii, ix, 8, 11, 25, 29, 38, 49, 50, 60, 63, 87

**SVMs** Support Vector Machines. 4, 8, 21, 25

**VGG** Visual Geometry Group. 13

**WMWB** Western Mediterranean Wetland Birds. 14, 23, 32, 82

**ZCR** Zero-Crossing Rate. 9

# Contents

# Chapter 1

# Introduction

The introduction outlines the research on classifying bird species through their audio calls, starting with the background on audio classification in ornithology and conservation biology. It presents the problem statement, emphasising the challenges of automated identification, and defines the study's objectives and methodologies. The scope and limitations of the project will also be highlighted, which sets the stage for the detailed exploration of Machine Learning (ML) approaches in the subsequent sections.

## 1.1 Background to Study

From the moment we enter this world, our sense of hearing plays a crucial role in guiding our interactions, and helping us appreciate the diverse collection of sounds that surround us. As our world becomes increasingly interconnected and automated, scientists and engineers have been driven by a compelling vision: to create machines that can perceive and understand sound as intuitively as humans do. This ambition has given rise to the field of audio classification, a journey that mirrors our own cognitive processes of listening, interpreting, and categorising the sounds we hear.

One of the most exciting frontiers for audio classification lies in environmental monitoring, particularly in bioacoustics—the study of animal sounds in nature. Bird species identification stands out as a prominent and challenging application of bioacoustics. Birds, with their diverse and often complex vocalisations, serve as excellent indicators of environmental change. However, accurately identifying bird species from audio recordings

presents several unique challenges. Individual variations in bird calls, the presence of background noise, and the need for high-quality training data all pose significant hurdles [1]. Despite this, the potential benefits of automated bird species identification are immense. From aiding in large-scale biodiversity surveys to enabling real-time monitoring of threatened species, this technology promises to revolutionise our understanding and conservation of avian populations.

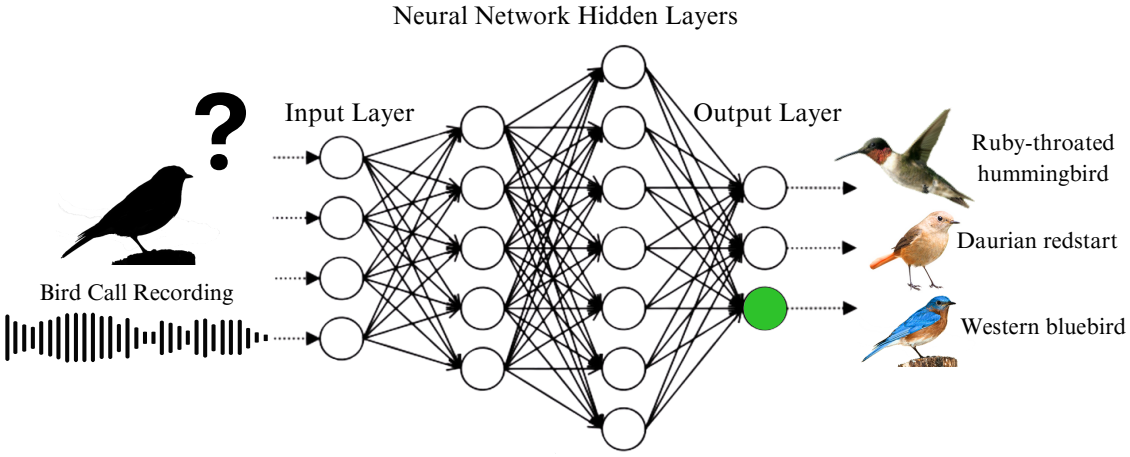



Figure 1.1: Overview of bioacoustic classification [2–4]

It is at this intersection of cutting-edge technology and pressing environmental needs that this research is positioned. By exploring and comparing the efficacy of various ML models in bird species classification based on audio recordings, this project aims to contribute to the ongoing efforts to refine and enhance this crucial tool for environmental monitoring and conservation.

## 1.2 Problem Statement

The advent of ML has opened new avenues for automating the bird species classification process, promising increased efficiency and scalability in biodiversity monitoring and conservation efforts. Although existing studies have shown promising results [5–7], several critical gaps in our knowledge persist:

- There is a significant discrepancy between the performance of these models in controlled, curated datasets and their effectiveness in real-world conditions.

- The presence of background noise and varying recording conditions often lead to a marked degradation in classification accuracy.
- The inherent variability in vocalisations within a single species – influenced by factors such as geographic location, individual variation, and context – poses a unique hurdle for accurate classification.

Addressing these knowledge gaps is crucial for several reasons. Accurate, automated bird species classification has far-reaching implications for biodiversity monitoring, enabling more comprehensive and frequent assessments of ecosystem health and the impacts of environmental changes. [6, 8]. As discussed by [9], it is an invaluable tool for conservation efforts, and facilitating the identification and tracking of endangered or threatened species, which can inform targeted protection strategies. Furthermore, in the context of ecological research, such technology can dramatically enhance our understanding of avian behaviour, communication patterns, and broader ecological dynamics.

To address these challenges and advance the field, this project aims to conduct a comprehensive survey of ML approaches to bird species classification from audio recordings. By synthesising current knowledge and highlighting areas for further investigation, the project seeks to contribute to ongoing efforts to create more accurate, robust, and widely applicable bird species classification systems, ultimately supporting critical conservation and ecological research initiatives.

## 1.3 Objectives

The broad objective of this research project is to advance the field of automated bird species identification using ML techniques, ultimately supporting biodiversity monitoring and ecological research. To achieve this, the project focuses on the following specific objectives:

- Conducting a comprehensive review of existing research on bird species classification using ML techniques. This will provide a solid foundation for this study and identify gaps in current knowledge.
- Investigating audio feature extraction methods to analyse their impact on classification accuracy. This is crucial for optimising the input data for the ML models,

potentially improving overall performance.

- Developing strategies to address vocalisation variations within species. This objective is essential for creating robust models that can handle the natural diversity in bird calls and songs.
- Conducting a thorough evaluation of traditional ML approaches and advanced Deep Learning (DL) architectures. This comparison will help identify the most effective techniques for bird species classification.

By addressing the outlined objectives, this study aims to provide a comprehensive understanding of the current state and future potential of ML in this field. The insights gained will contribute to the development of more accurate, robust, and practical automated systems, ultimately supporting effective biodiversity monitoring and advancing ecological research.

## 1.4 Scope and Limitations

### 1.4.1 Scope of the Project

The defined scope enables a thorough exploration of ML techniques in bird species classification, covering a diverse array of models and a significant number of species. This approach is structured to yield valuable insights into the effectiveness of various algorithms and feature extraction methods in this field.

- This project aims to classify a minimum of 20 distinct bird species, carefully selected to represent a wide variety of vocalisations, habitats, and geographical distributions.
- The research will assess multiple ML models, including both traditional algorithms and DL architectures. The traditional algorithms will include Support Vector Machines (SVMs), Random Forests (RFs), XGBoost, and k-Nearest Neighbours (KNNs). The DL architectures will include Convolutional Neural Networks (CNNs), and Feedforward Neural Networks (FFNNs).
- The system architecture will be designed to allow for future expansion, including the addition of new species and integration of larger datasets.

### 1.4.2 Limitations of the Project

While striving for a robust and effective classification system, several constraints and limitations must be acknowledged:

- Due to time constraints, the project is restricted to 20 bird species, which may not fully capture the vast diversity of avian vocalisations globally. The quality and quantity of audio recordings may differ across species, potentially introducing biases in model training and evaluation. Rare species or those with limited vocalisation data may be underrepresented.
- The complexity and scale of DL models may be constrained by the available computational resources, limiting the extent of hyperparameter tuning and model size.
- A limited budget (R2000) may restrict access to certain proprietary datasets or high-performance computing resources.

## 1.5 Thesis Outline

The remainder of this report is organised as follows:

**Chapter 2, Literature Review:** This chapter presents a comprehensive overview of existing research in bird species classification using machine learning, identifying key trends and gaps in current knowledge.

**Chapter 3, Project Methodology and Design:** This chapter outlines a structured overview of the methodologies used in the project and the key design choices.

**Chapter 4, Project Implementation:** This chapter describes the implementation phase, elaborating on the practical steps taken to execute the design.

**Chapter 5, Results:** This chapter produces a comprehensive analysis of the results.

**Chapter 6, Conclusion and Future Recommendations:** This chapter summarises the key findings of the project, evaluates the outcomes against the original objectives, discusses the limitations encountered, and highlights what worked well in the study.

# Chapter 2

# Literature Review

## 2.1 Introduction

This literature review explores the rapidly evolving field of bird species classification, focusing on the intersection of bioacoustics, ML, and conservation biology. This chapter examines the journey from traditional classification methods to advanced DL techniques, emphasising the transformative impact of artificial intelligence on bioacoustic analysis. It explores the bird call datasets available to researchers, the persistent challenges in classification, and the innovative approaches being developed to overcome these obstacles.

The review also considers the broader implications of these advancements for avian conservation efforts. Finally, it looks ahead to emerging trends and future directions in bioacoustic monitoring that promise to reshape our understanding and protection of avian biodiversity. Through this comprehensive exploration, the literature review aims to provide a strong foundation for understanding the current state of bird species classification models and its potential to revolutionise ecological research and conservation practices.

## 2.2 Evolution of Bird Species Classification

This section traces the progression of approaches used to identify and categorise bird species, from traditional methods relying on human expertise to the cutting-edge computational techniques employed today.

### 2.2.1 Traditional Methods in Bird Species Identification

In the preliminary stages, bird sound classification relied heavily on the expertise of ornithologists who could identify species by ear. While effective for small-scale studies, this method was time-consuming, subject to human judgment, and challenging to scale for large datasets or real-time monitoring [5]. The introduction of audio recording technology marked a significant advancement, enabling the development of spectrograms—visual representations of sound frequencies over time (see Fig. 2.1). Experts could visually analyse these spectrograms to identify patterns and characteristics specific to different bird calls [6]. However, manual inspection of spectrograms remained a laborious and specialised task, susceptible to human error and bias [7].

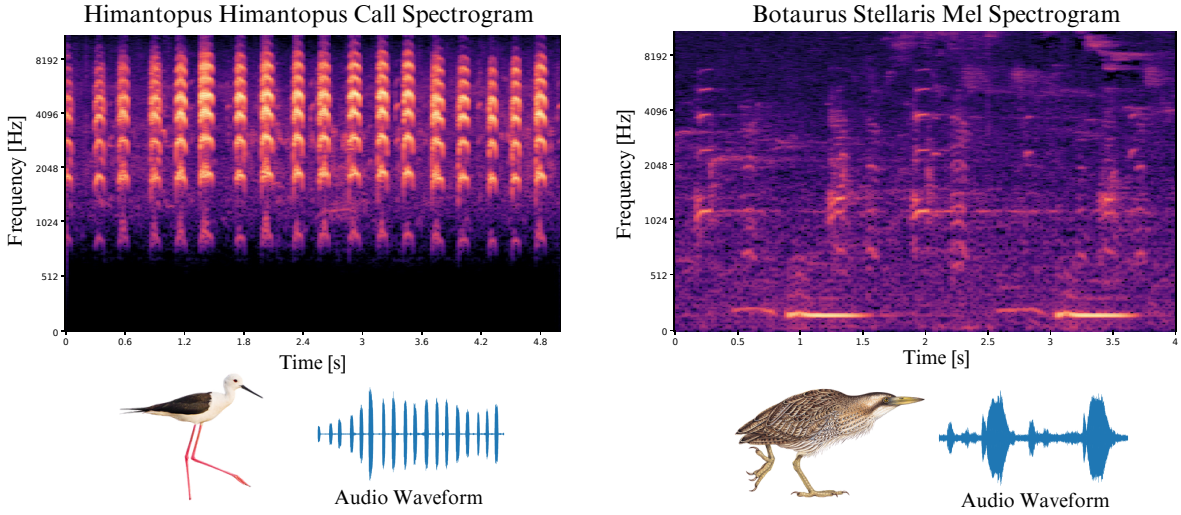



Figure 2.1: Spectrograms of different birds [10, 11]

As digital signal processing techniques advanced, researchers began exploring automated approaches to bird sound classification. These early automated methods often involved extracting handcrafted acoustic features from audio recordings and employing traditional supervised learning algorithms for classification. In this context, supervised learning involves training a model on a dataset of bird sounds with known species labels. The algorithm learns to associate specific acoustic features with particular bird species. Once trained, the model can then be applied to unseen, unlabeled bird sound recordings to predict the species. Key methods included:

1. **Random Forests (RFs)**: This is a ensemble supervised ML technique used for

classification and operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes considered [12]. This learning method employs Bootstrap Aggregating, which involves creating multiple subsets of the training data through random sampling with replacement [13]. RFs have been successfully used for classifying 20 bird species and achieved an average accuracy of 95.35% [6].

2. **XGBoost**: This is an ensemble learning method and works on a similar principle to RFs by leveraging the power of multiple decision trees to make predictions [1]. It works by boosting, where each tree is built to correct the errors made by the previous trees. With large datasets, training can be computationally expensive, but it often outperforms RFs in terms of accuracy [1, 14].

3. **Support Vector Machines (SVMs)**: Known for their effectiveness in handling high-dimensional data, SVMs were used in conjunction with various acoustic features to classify bird species based on their vocal characteristics [15, 16]. One study achieved 100% accuracy in classifying 11 bird species using an SVM model which used MFCCs as acoustic features. SVMs offered improved accuracy compared to earlier methods but, like other traditional ML approaches, still relied on handcrafted features, limiting their ability to capture the full complexity of bird sounds [6, 15].

4. **k-Nearest Neighbours (KNNs)**: KNN is an algorithm that classifies data points by identifying the majority class among their closest neighbours in the feature space. Reference [17] highlights the use of KNN in bird classification to compare with more complex models. This methods struggle with the variability, scalability, and complexity of bird vocalisations, especially in noisy real-world conditions [15, 18].

5. **Other Traditional ML Methods**: Researchers also explored a range of algorithms, including Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) [7, 12, 15]. GMMs model data as a mixture of Gaussian distributions, while HMMs use probabilities to model transitions between hidden states and the generation of observable outputs. A study combined Mel-Frequency Cepstral Coefficients (MFCCs) and short-time energy with a dual Gaussian Mixture Model to classify eight bird species, achieving an average recognition rate of 94.35% [7].

### 2.2.2 Advanced Feature Extraction in Bioacoustic Analysis

Feature extraction techniques play a crucial role in bioacoustics, enabling researchers to transform raw audio data into meaningful representations that facilitate accurate analysis and classification of animal vocalisations.

Frequency and time-frequency audio representations are fundamental to many bioacoustic analyses. MFCCs are a widely used feature extracted from Mel-spectrograms (see Fig. 2.1), which employ the Mel scale to align frequency with human auditory perception. The MFCCs provide a compact representation of the spectral envelope [7, 17, 19, 20]. Their robustness to variations in recording conditions makes them particularly useful in field recordings [19, 20]. Chroma features, which represent pitch content by dividing the frequency spectrum into 12 semitone bins, are useful for identifying sounds with strong harmonic structures, such as many bird songs [13].

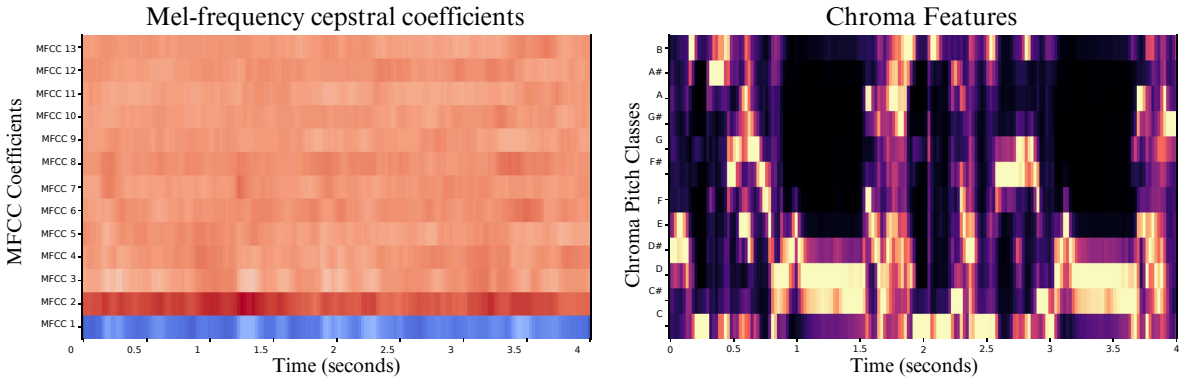



Figure 2.2: Examples of MFCCs and chroma features

The Constant-Q Transform (CQT) is a well-regarded algorithm in signal processing, widely used to compute the Fourier Transform and analyse the frequency content of non-periodic signal. It is a powerful tool used in various audio analysis applications and has been used to prepare data for Deep Learning (DL) models to classify bird species [21].

Time-Domain Features are often used by researchers because they offer insights into the temporal characteristics of sound signals. Root Mean Square (RMS) measures average energy, helping to identify high-energy sounds [13, 22]. The Band Energy Ratio (BER) compares energy in different frequency bands, aiding in the identification of sounds with particular spectral shapes [22]. The Zero-Crossing Rate (ZCR) is valuable for identifying highly periodic sounds like certain vocalisations [9].

### 2.2.3 Deep Learning (DL) in Bird Vocalisation Classification

DL models have revolutionised audio classification, enabling more accurate, efficient, and scalable analysis of audio data compared to traditional methods.

#### Convolutional Neural Networks (CNNs)

CNNs have emerged as a dominant architecture in audio classification, particularly excelling at extracting spatial features from spectrograms. Dating back to 1998, their strength lies in their ability to detect patterns and features within local regions of an image by hierarchical feature learning [6, 18, 20]. CNNs have emerged as the dominant approach for audio classification tasks, consistently demonstrating superior performance across several studies [7, 15, 18, 23]. Fig 2.3 below shows a CNN sequence breakdown:

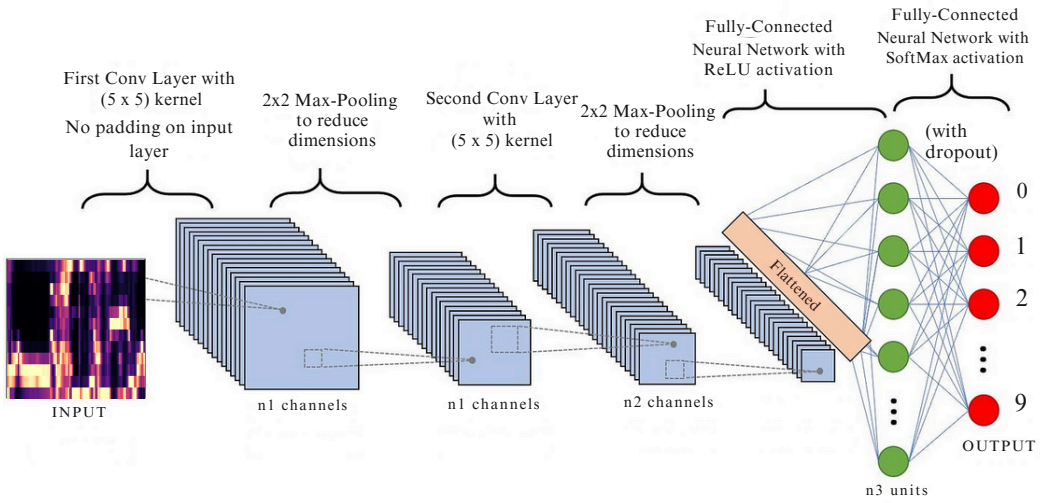



Figure 2.3: CNN sequence overview [24]

#### Recurrent Neural Networks (RNNs)

RNNs are specifically designed to process sequential data, making them well-suited for handling the temporal dependencies inherent in audio signals. Their key strength lies in their ability to maintain a memory of past inputs, allowing them to capture temporal relationships within audio sequences [15]. They can suffer from vanishing or exploding gradients during training, especially when dealing with long sequences, and their training can be computationally expensive when struggling to find optimal parameters [15]. This

is because they have weights and biases shared between inputs. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been developed to avoid the exploding/vanishing problem by using separate paths to make future predictions [15]. GRUs have a simpler structure and fewer parameters compared to LSTMs. This can make them computationally less expensive and faster to train [25]. Few studies have investigated the use of RNNs for bird species classification tasks. In cases where they are utilised, hybrid models are often developed, such as the GRU-CNN (CRNN) model which produced a validation accuracy of 50.17% [25].

### Transformers

Transformers have recently gained traction in audio classification due to their ability to capture long-range dependencies and handle sequential data more effectively than traditional RNNs, thanks to their attention mechanism [15, 26]. In bird sound recognition, models like Transound, a hyper-head attention transformer, have shown significant promise in achieving high accuracy rates. However, transformers come with a higher computational cost for training, as they require large datasets, making them more resource-intensive compared to Convolutional Neural Networks (CNNs) [26].

### Hybrid Models

Hybrid models leverage the strengths of different DL architectures to address the limitations of individual models and achieve enhanced performance in audio classification. CNN-RNN hybrids leverage the spatial feature extraction capabilities of CNNs and the temporal dependency handling of RNNs. CNN-Transformer hybrids combine the spatial feature extraction of CNNs with the long-range dependency modelling capabilities of transformers. Some hybrid models integrate DL models with traditional ML classifiers, such as using a CNN for feature extraction and feeding the extracted features into a SVM or a RF classifier [15, 17].

One study investigated a hybrid modeling strategy that leveraged the capabilities of CNNs and RNNs for classifying bird species [27]. The proposed method utilized a CNN model that took Mel-frequency and Short Term Fourier Transform spectrograms as inputs. The features extracted by the CNN were then fed into various RNN variants, such as LSTM and GRU networks with Legendre Memory Units (LMU). The hybrid models

achieved an average accuracy of 67%, with the highest accuracy reaching 90%

### 2.2.4 Leveraging Transfer Learning for Improved Audio Classification

Transfer learning has emerged as a powerful technique in audio classification, particularly in the field of bird call identification. This approach allows models to leverage knowledge gained from large, diverse datasets and apply it to more specific tasks, addressing the challenge of limited labelled data in many bioacoustic studies. The core principle of transfer learning in bird call classification involves pre-training models on extensive audio datasets, such as AudioSet or BirdNET's training data, to learn general acoustic features relevant across various bird species and environments [1,7,18]. These pre-trained models are then fine-tuned on smaller datasets, allowing them to adapt to particular characteristics while retaining the broader knowledge acquired during pre-training [5, 7, 18]. This means they can perform as well as or better than complex architectures used in other domains [7].

Prominent examples of transfer learning in bird call classification include BirdNET (see Fig. 2.4) and Alex adaptation.

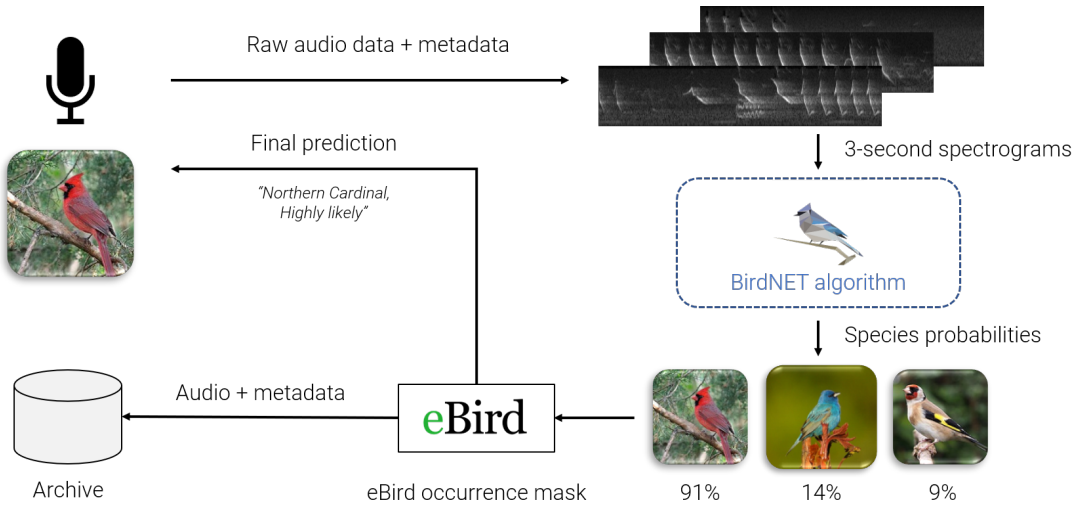



Figure 2.4: BirdNET workflow diagram [28]

In addition to these transfer learning approaches, the field of bioacoustic analysis has seen the development of comprehensive tools for automated biodiversity monitoring. A notable example is the Automated Remote Biodiversity Monitoring Network (ARBIMON),

which offers a comprehensive approach to acoustic data analysis and species identification. The system allows for real-time monitoring and employs ML algorithms, particularly Hidden Markov Model (HMM), for automated species identification. ARBIMON provides access to a vast database of recordings, exceeding 1.3 million 1-minute samples as of 2013, offering invaluable resources for bioacoustic research and monitoring efforts [19, 29].

CNNs appear across multiple studies in various architectural pre-trained variations. VGG is known for its simplicity and is often a baseline for comparison with more complex models [9, 30, 31]. It was developed to increase the depth of CNNs and has 16- and 19-convolutional-layer variants. It was trained on the ImageNet dataset, which consists of over 14 million images belonging to an estimated 1000 classes.

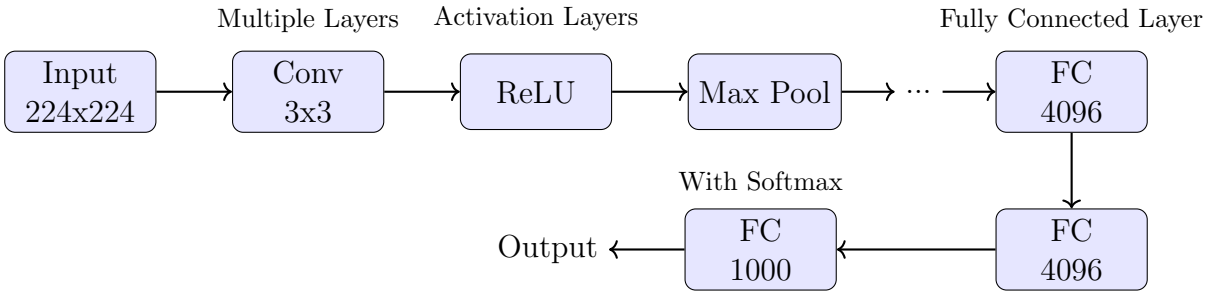



Figure 2.5: VGG architecture (left-to-right)

ResNet is another CNN architecture with the ability to learn very deep networks through residual connections trained on the ImageNet dataset [7, 9, 18, 32]. It has 8 times more layers than VGG, while having lower complexity. BirdNET, a widely used bird sound identification system, uses a ResNet architecture to classify over 984 North American and European bird species [7, 26]. EfficientNet is similar to the above, but offers a good balance between model size and performance, which provides efficient classification with fewer computational resources [33]. It was also trained on the ImageNet dataset and its B-7 variant achieved an accuracy of 84.3% at a relatively faster speed than its competition.

## 2.3 Bioacoustic Data Resources and Challenges

This section explores the various resources available for obtaining bird vocalisation recordings and the inherent challenges in assembling comprehensive datasets.

### 2.3.1 Comprehensive Bird Vocalisation Datasets

Xeno-canto [34] is a widely used community-driven platform hosting a vast collection of bird sound recordings contributed by users worldwide [5,7,35]. It encompasses recordings from various geographical locations, covering a wide range of bird species, and is often the source of audio samples for various datasets.

BirdCLEF is an annual competition challenging researchers to develop innovative bird sound recognition algorithms [1, 9, 18]. BirdCLEF datasets often consist of soundscape recordings, capturing a mixture of bird vocalisations and environmental sounds that are verified by expert ornithologists. This complexity reflects real-world monitoring scenarios and pushes the boundaries of classification algorithms.

The Macaulay Library of Natural Sounds, part of the Cornell Lab of Ornithology, houses one of the world's largest collections of animal sounds, including a vast repository of bird vocalisations [7, 36]. Each recording is accompanied by detailed metadata, such as species identification, location, date, and recording conditions. The Macaulay Library has a rigorous quality control process, with recordings reviewed and annotated by experts to ensure accuracy. This focus on data quality makes it a valuable resource for training and evaluating bioacoustic models.

The Western Mediterranean Wetland Birds (WMWB) Dataset focuses specifically on bird species inhabiting wetlands in the Western Mediterranean region, comprising annotated excerpts of vocalisations from 20 endemic bird species. The dataset provides hard labels, indicating the precise start and end times of each bird vocalisation, enabling detailed analysis of acoustic events [37].

### 2.3.2 Obstacles in Accurate Bird Call Classification

Bird call classification faces several significant challenges that stem from the complex nature of avian vocalisations, the diverse environments in which they are recorded, and the availability of quality audio samples in datasets.

#### Availability of Quality Datasets

Many bird call datasets suffer from class imbalance, with some species significantly over-represented compared to others. This imbalance can bias models towards the majority classes, leading to poor performance on under-represented species [1, 6, 14, 18].

Some datasets provide only weak labels, indicating the presence of a species in a recording without specifying the temporal location of the vocalisation [9, 38]. This lack of temporal information can limit the development of models for tasks requiring precise call detection or analysis of temporal patterns within vocalisations. Models trained on weakly labelled data may struggle to pinpoint the exact timing of calls, hindering their performance and increasing loss during training [25, 38].

#### Availability of Quality Audio Samples

One of the primary challenges is the presence of background noise in recordings. Natural environments inevitably contain sounds such as wind, rain, traffic, and other animal vocalisations, which can mask or distort bird calls [16, 37]. Multiple birds often vocalise simultaneously, creating a complex acoustic environment where individual calls are difficult to isolate [1, 7, 37, 39]. This noise significantly impacts the accuracy of classification algorithms, often leading to increased false positive detections [18]. Researchers found that it is crucial to strike a balance between noise reduction and signal preservation. Low signal-to-noise ratios, particularly due to distance or ambient sounds, also present a substantial challenge in soundscape recordings [7].

The inherent variation in bird calls themselves presents yet another challenge. Bird calls can exhibit significant differences across individuals, species, and geographic regions [6, 39, 40]. This variability requires classification models to generalise across a wide range of call variations to accurately identify bird species [39].

### 2.3.3 Solutions to Classification Challenges

To combat the challenge of sample imbalance in datasets, researchers often oversample the minority by duplicating the samples and performing data augmentation techniques [13, 41]. This is a way to provide acceptable data and they play a crucial role in bird call classification in addressing the challenges above and enhancing model robustness.

Common audio augmentation techniques, such as time stretching and pitch shifting, are widely used to replicate the natural variability of bird calls [7, 41]. Fig. 2.6 highlights additional effective methods, including the introduction of background noise and blending multiple bird call recordings to generate more complex training samples [7, 18].

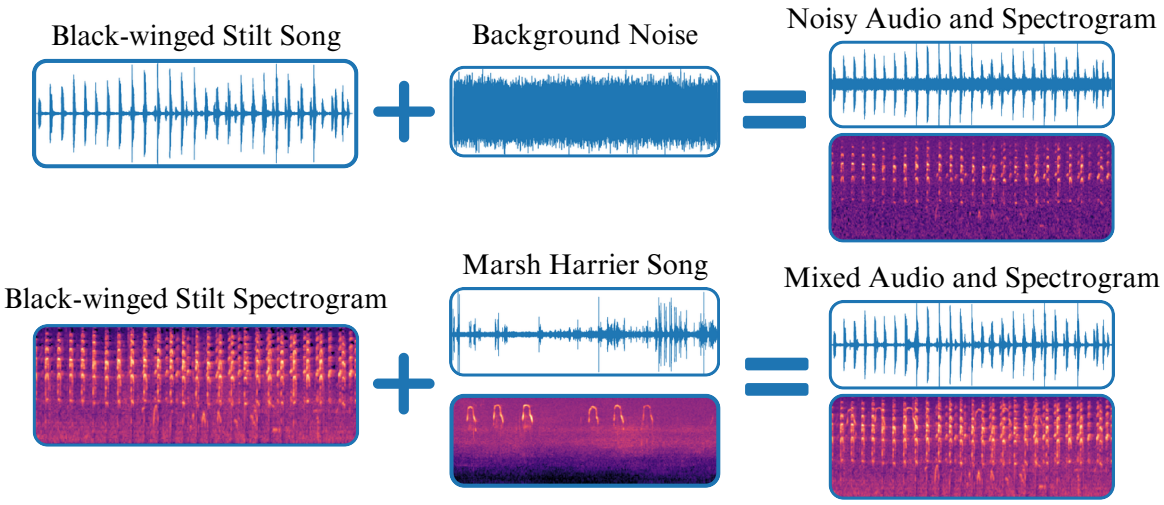



Figure 2.6: Audio augmentation techniques

These techniques effectively tackle the issue of limited data in bioacoustics by synthetically expanding dataset size and diversity, thus preventing overfitting [15, 41, 42]. They have also shown to significantly improve model robustness by exposing models to a wider range of acoustic conditions during training [7, 15, 18].

Mindlin and Tubaro [41] developed an innovative approach to data augmentation after recognising that existing methods, primarily derived from image processing techniques, were inadequate for replicating the complexity of authentic bird vocalisations. Their approach is based on the biomechanical mechanisms of birdsong production to vocalise syallables. They created a dynamic system that simulates the oscillations of the bird's vocal muscle control to create synthetic samples. They train a CNN on the synthetic data and achieved an average F1-score of 81%.

Species-specific models, tailored to particular species or groups of similar species, can also account for variations in calls by focusing on the subtle distinctions between closely related species [40]. Researchers also consider selecting appropriate evaluation metrics that account for class imbalance or other biases in the data, such as F1-score, area under the ROC curve (AUC), or precision and recall [35].

## 2.4 ML's Role in Advancing Bioacoustics and Conservation

ML has significantly transformed bioacoustics and bird conservation efforts. The technologies have noticeably improved the accuracy and efficiency of bird surveys and have demonstrated the capability to surpass human performance in certain contexts [16, 23, 43]. This enabled large-scale monitoring programmes and provided valuable insights into avian ecology for conservation planning [7, 19, 40].

A study highlighted the effectiveness of a CNN-based bird sound classifier in identifying species such as the Grey Vireo, outperforming other recognisers across various accuracy metrics [40]. Another research project highlighted the successful application of CNNs in classifying northern spotted owl calls and other forest bird species, demonstrating its utility in monitoring sensitive populations and their habitats [43]. The development of AMResNet, a model integrating ResNet with attention layers, further exemplifies the progress in bird sound classification using advanced DL architectures [17].

ML applications in bioacoustics have also facilitated research into the impacts of climate change on bird populations [7, 40] with tools like BirdCLEF. With its user-friendly application, BirdCLEF has empowered the public to actively participate in bird monitoring efforts. This engagement has led to the creation of extensive datasets, which are invaluable for training and validating ML models [18, 40]. The surge in data facilitated by citizen scientists has enabled researchers to conduct broader and more comprehensive studies, significantly expanding the scope and depth of ornithological research [44].

Moreover, ML has been instrumental in studying the impact of noise pollution on birds, particularly in urban environments. A study focusing on bird songs in urban areas found that birds tend to sing at higher frequencies in cities, likely to avoid overlap with low-frequency anthropogenic noise. This research exemplifies how ML can be applied to analyse acoustic data, revealing behavioural adaptations of birds in response to urban noise [45].

## 2.5 Future Prospects in Bioacoustic Monitoring

The field of bioacoustic monitoring is rapidly evolving, with several emerging trends and future directions shaping its development. One significant area of research is unsupervised learning, which aims to discover patterns and structures in data without relying on pre-existing labels [38, 46]. However, reference [38] found that assessing the performance of unsupervised models can be more challenging than supervised models, as there are no pre-defined labels to measure accuracy. Understanding the patterns discovered by unsupervised models can be complex, requiring careful analysis and domain expertise.

Real-time bioacoustic monitoring is another emerging trend, with growing demand for systems capable of processing data in the field, providing immediate insights and enabling rapid responses to ecological events [19, 29]. This development is closely tied to the integration of bioacoustic data with other environmental data sources, such as weather patterns, satellite imagery, or habitat characteristics, which can provide a more comprehensive understanding of ecosystem dynamics [44]. Integrating bioacoustic sensors with other sensors (e.g., camera traps, weather stations) can create more powerful and informative monitoring systems, enabling the study of species interactions, behavioural responses to environmental changes, and the detection of complex ecological events. Due to ethical concerns, it is crucial to ensure responsible usage, and environmental disturbances [18].

## 2.6 Chapter Summary

In conclusion, the integration of ML techniques in bioacoustics has significantly advanced our understanding of avian ecology and conservation efforts. By automating the identification of bird species from audio recordings, ML models have not only improved the accuracy and efficiency of bird surveys but have also surpassed traditional manual methods in various contexts [23, 40]. The successful application of DL architectures, such as CNNs and novel models like AMResNet, highlights the potential of these technologies to monitor sensitive populations and their habitats effectively [17, 43].

The field still faces challenges in standardisation and evaluation, making it difficult to compare results across studies and to draw meaningful conclusions about the relative

performance of different methods [23]. There is a growing need for larger-scale comparisons involving a wider range of species, datasets, and acoustic environments to provide a more robust assessment of the generalisability and limitations of different approaches [23]. While many studies have demonstrated the potential of bioacoustic monitoring techniques in controlled settings, there is a need for more research evaluating performance in real-world scenarios with complex soundscapes, varying recording conditions, and overlapping vocalisations [7, 18].

Ultimately, the future of bioacoustic monitoring lies in the collaboration between advanced ML techniques and comprehensive ecological frameworks, facilitating a deeper understanding of the intricate relationships within ecosystems, especially in the context of changing environmental conditions and anthropogenic impacts [7, 12, 19].

# Chapter 3

# Project Methodology and Design

This section outlines the foundational architecture and key design considerations that guide the development of the bird species classification system. The methodology for this project is structured to facilitate a systematic and rigorous approach to bird species classification using audio recordings by integrating best practices from the literature and employing a well-defined framework. The overview of the methodological approach is shown in Fig. 3.1.

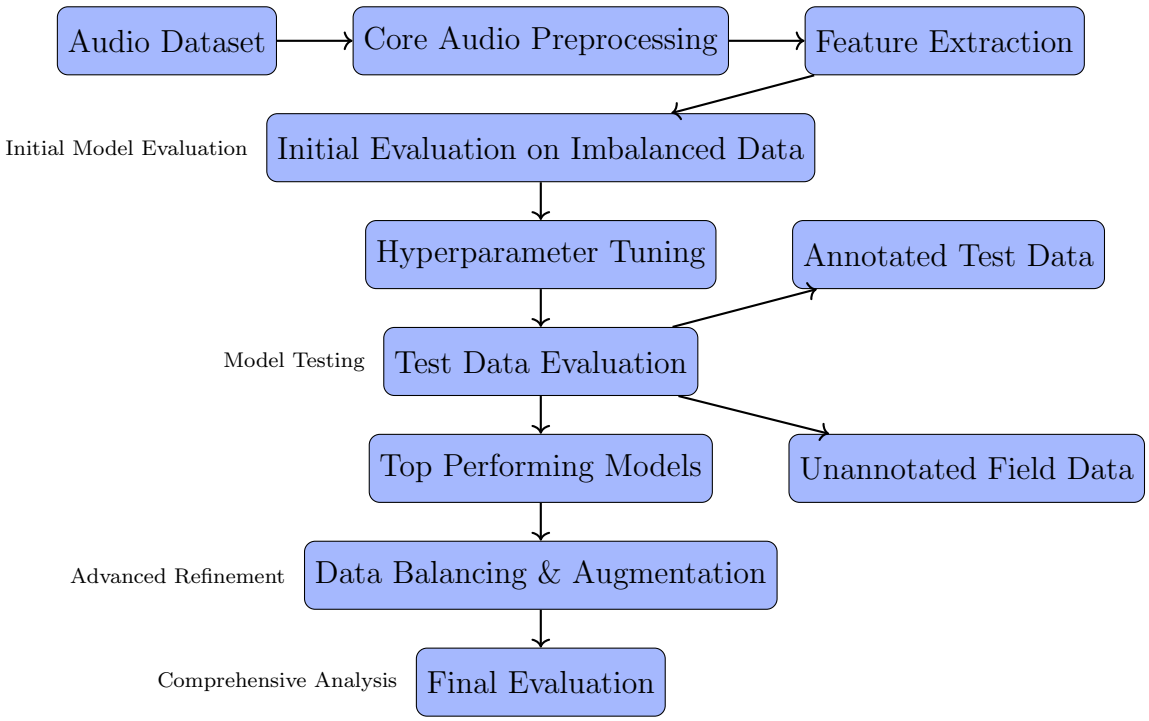



Figure 3.1: Research methodology overview

## 3.1 Project Requirements and Specifications

The project requirements, and specifications are highlighted in Tab. 3.1, and 3.2.

| R-ID | Bird Species Classification Project Requirements | S-ID |
|---|---|---|
| R-01 | Obtain a comprehensive dataset of bird calls representing multiple bird species with annotated bird calls. | S-01, S-02, |
| R-02 | Develop a reproducible and scalable software pipeline to extract relevant audio features from bird call recordings. | S-03 |
| R-03 | Implement both traditional ML and DL models for bird species classification. | S-04, S-05, S-06, S-08, S-09 |
| R-04 | Create a framework to assess and compare model performance. | S-07 |
| R-05 | Design a protocol for testing models in practical, field-like conditions. | S-10 |
| R-06 | Maintain detailed records of all processes, decisions, and results. | S-11 |

Table 3.1: Project requirements

| S-ID | Bird Species Classification Project Specifications | A-ID |
|---|---|---|
| S-01 | The dataset to be used should include a minimum of 2000 annotated field-like bird audio excerpts of at least 20 unique bird species. | A-01 |
| S-02 | The audio recordings should be either in MP3, WAV, or FLAC format, with a sample rate greater than 16 kHz. | A-01 |
| S-03 | Develop a pipeline that extracts key features from the audio, focusing on characteristics that represent both spectral and temporal aspects of the bird calls. | A-02, A-03, A-04 |
| S-04 | Implement and train models such as SVMs, and KNNs to serve as benchmarks for comparison with deep learning models. | A-05, A-06 |
| S-05 | Design and train neural network architectures that are effective in audio classification tasks. | A-05, A-06 |
| S-06 | Incorporate transfer learning by leveraging pre-trained models on similar tasks to improve performance. | A-05, A-06 |
| S-07 | Measure the accuracy, precision, recall, and F1-score of each model, alongside other performance indicators like model complexity, training time, and inference speed. | A-07 |
| S-08 | Obtain a minimum accuracy of 70% for at least one traditional algorithm. | A-08 |
| S-09 | Obtain a minimum accuracy of 80% for at least one DL architecture. | A-09 |
| S-10 | Conduct evaluations using field recordings not included in the training data and obtain a minimum accuracy of 60%. | A-10 |
| S-11 | Ensure that all the code is well documented and reproducible for feature extraction, model training, and performance evaluation. | A-11, A-12 |

Table 3.2: Project specifications

| A-ID | Bird Species Classification Project ATPs |
|---|---|
| A-01 | Confirm the dataset meets the diversity and quality requirements. |
| A-02 | Verify that any data augmentation techniques, if used, are correctly applied to ensure variability without data distortion. |
| A-03 | Validate the accuracy of feature extraction against a reference implementation. |
| A-04 | Conduct visual inspections of extracted features to ensure their integrity and consistency across the dataset. |
| A-05 | Verify the proper implementation of both traditional and deep learning models. |
| A-06 | Ensure that the model architectures adhere to design specifications and that the data is properly split into training, validation, and test sets. |
| A-07 | Validate the accuracy of performance metrics, including accuracy, precision, recall, and F1-score. |
| A-08 | Validate that the minimum accuracy of 70% for a traditional model has been obtained. |
| A-09 | Validate that the minimum accuracy of 80% for a DL architecture has been obtained. |
| A-10 | Verify that the minimum accuracy of models that are tested under realistic, and challenging environments is 60%. |
| A-11 | Review the documentation to ensure comprehensive coverage of methodologies, implementation details, and results. |
| A-12 | Verify that the entire workflow is reproducible, ensuring that all code is version-controlled and accessible. |
| A-13 | Verify that no part of the project violates ethical guidelines related to environmental conservation or research practices. |

Table 3.3: Project ATPs

The success of this research project hinges on a well-defined set of requirements and specifications that guide its development from data acquisition to model implementation and performance evaluation. Each requirement and specification is designed to facilitate effective model testing and validation, ensuring that the final models not only perform well in controlled settings but are also robust enough for real-world application.

## 3.2 Project Design Choices

The design choices made in this project reflect a deliberate and thoughtful approach to achieving accurate bird species classification from audio recordings. These decisions are informed by the literature review, where specific challenges associated with processing natural soundscapes, the need for robust feature extraction methods, and the comparative strengths of various ML models are explored.

### 3.2.1 Dataset Selection

To aid this project, two datasets are examined: the BirdCLEF 2023 dataset and the WMWB dataset. The selection process was guided by the project's requirement for high-quality, annotated data, as well as real-world examples to support effective model evaluation. Ultimately, one dataset was chosen to align with these objectives.

The BirdCLEF 2023 dataset presents a challenging collection of real-world, non-annotated soundscape recordings that encapsulate the complexity of natural environments. In these recordings, individual bird calls are not labeled, which may complicate the training process. The absence of annotated data poses a significant challenge for the model's ability to accurately capture the nuances of different bird calls. Consequently, this dataset does not align with the project's objectives.

On the other hand, the WMWB dataset offers high-quality annotated data, with precise start and end times for each bird vocalisation. This dataset features 20 bird species, which meets the project's requirements. The audio is in MP3 format, with sample rates ranging from 22 kHz to 48 kHz—more than adequate for capturing the relevant acoustic features. The WMWB dataset allows the model to train on clean, well-defined bird calls, providing a controlled environment for evaluating model performance under ideal conditions. This is the dataset that will be explored for this task.

### 3.2.2 Audio Preprocessing Techniques

From the literature review, it was often emphasised that good preprocessing of the data significantly improves the model's performance. The techniques implemented in this study include windowing, and padding with pink noise as outlined below.

- **Windowing:** Windowing, or audio segmentation, ensures that features extracted from audio samples have consistent time dimensions, preventing the need to stretch or compress data, which can degrade quality and introduce distortion.
- **Padding with Pink (1/f) Noise:** Pink noise, which simulates natural background sounds, is used to pad shorter audio samples to reach the predefined window length. It is a more realistic augmentation choice than white noise, as it better resembles the environment where the recordings were made.

Frequency filtering was initially considered as a potential noise reduction technique. However, after careful analysis of the diverse bird vocalisations within the chosen dataset, it became apparent that there was no universally effective method to mitigate noise without potentially compromising the model's performance. Whilst natural background noise often occupies the lower frequency ranges, and its removal would typically be beneficial, a notable exception was observed. The *Botaurus stellaris* (Eurasian bittern) produces a remarkably low-pitched vocalisation that dominates the lower frequencies (see Fig. 2.1). This discovery rendered the application of a broad low-frequency filter ineffective, as it would risk eliminating crucial species-specific acoustic information.

### 3.2.3 Feature Extraction Methods

The following feature extraction methods will be used for the bird classification project:

| Method | Rationale |
|---|---|
| MFCCs | MFCCs are a widely adopted feature representation in bioacoustic studies and they are robust to noise and recording conditions. This makes them ideal for field recordings and suitable for capturing subtle variations in bird calls for classification tasks. |
| Mel-spectrogram | The Mel-spectrogram, which underpins the calculation of MFCCs, will also be used as a feature representation. This method offers a detailed view of how bird calls change over time, enhancing temporal and spatial analysis. |
| Chroma features | This feature can complement the spectral envelope information provided by MFCCs, potentially improving the model's ability to capture the nuances of bird vocalisations. |
| CQT | CQT provides better frequency resolution at lower frequencies and lower resolution at higher frequencies, which aligns well with how bird vocalisations might vary across different species. |

Table 3.4: Feature extraction methods and rationale

While time-domain features such as RMS power can offer insights into the temporal characteristics of sound signals, these features will not be used in this project. The reason for this decision is that the audio recordings in the dataset may have varying and inconsistent volume levels, as well as unpredictable zero-crossing patterns, which could diminish the reliability and effectiveness of these time-domain features for the task of bird species classification.

### 3.2.4 Model Selection

Considering the insights gained from the literature review, the following traditional models in Tab. table 3.5 will be investigated for the bird audio classification project:

| Model | Rationale |
|---|---|
| RF | RFs are a robust and versatile ensemble learning technique that have been successfully applied to bird species classification tasks. By constructing multiple decision trees and combining their outputs, RFs are less prone to overfitting compared to individual decision trees. The ability of RFs to capture non-linear relationships between the acoustic features and the target bird species makes them a promising candidate for this project. |
| XGBoost | As an advanced ensemble learning method, XGBoost has demonstrated superior performance compared to traditional Random Forests in several bird classification studies. The boosting approach, where each subsequent tree is trained to correct the errors made by the previous ones, can help the model effectively learn the intricate patterns within the bird audio data, potentially leading to improved classification accuracy. |
| SVM | SVMs have a strong track record in handling high-dimensional feature spaces, which is particularly relevant for the classification of bird vocalisations based on acoustic features such as MFCCs. The ability of SVMs to find optimal decision boundaries in the feature space can help differentiate between the unique vocal characteristics of various bird species, making them a suitable choice for this project. |
| KNN | KNNs is a simple, intuitive algorithm that has been used as a baseline in several bird classification studies. The algorithm can provide insights into the inherent separability of the bird species based on the selected acoustic features, helping to guide the development of more advanced classification models. |

Table 3.5: Traditional Machine Learning Models and Their Justifications

Based on an extensive literature review of deep learning algorithms used for bird species identification, a range of models has been selected for exploration. The selection aims to balance baseline performance, state-of-the-art techniques, and computational efficiency. The chosen models for this study are highlighted in Tab. 3.6.

While RNNs and Transformers are also prominent deep learning models in the audio classification domain, they have not been chosen for this project's exploration. RNNs, such as LSTM and GRU, have not shown consistently superior performance compared to CNNs for bird species classification tasks. Additionally, Transformers, despite their ability to capture long-range dependencies, tend to be more computationally expensive

| Model | Rationale |
|---|---|
| FFNN | As a baseline deep learning model, a FFNN will be implemented. By comparing the performance of FFNNs with traditional machine learning models, the potential benefits and limitations of deep learning approaches for this task can be better understood. |
| CNN | CNNs have emerged as a dominant architecture in audio classification, particularly for extracting spatial features from spectrograms. Their ability to hierarchically learn features from the input data makes them a promising choice for the bird audio classification task. Various CNN architectures will be explored to leverage their strengths in capturing the complex patterns within the audio recordings. |
| Transfer Learning | To further enhance the performance of the deep learning models, the project will explore the use of transfer learning by leveraging pre-trained CNN architectures, such as VGG, and ResNet. These models have been trained on large-scale datasets like ImageNet and have demonstrated strong feature extraction capabilities, which can be beneficial for the bird audio classification task, even though the original models were trained on image data. |

Table 3.6: Deep Learning Models and Their Justifications

and require larger datasets for effective training. Given the available resources and the project's focus, the computational complexity and data requirements of Transformers make them less suitable for the current investigation.

The comparative analysis of these deep learning models against the traditional machine learning techniques will contribute to a thorough understanding of the most effective approaches for classifying bird species based on audio recordings. To ensure a thorough evaluation, metrics such as accuracy, precision, recall, and the F1-score will be employed. These are essential for understanding how well the models handle the varying complexities in the data. For example, precision and recall will be particularly important in cases where the dataset exhibits class imbalance, as they provide insights into the models' ability to correctly classify bird species despite skewed distributions in the number of samples for each class. Additionally, the area under the ROC curve (AUC) will be used to further assess the models' ability to differentiate between classes.

### 3.2.5 Development Environment

Python is the primary programming language used in this study due to its extensive ecosystem tailored for machine learning, data science, and audio processing tasks. It

offers numerous libraries such as Librosa, Scikit-learn, TensorFlow, and NumPy, which are essential for audio processing, traditional machine learning, and deep learning tasks (see Appendix for an overview of their purposes and features). The language is clear and easy to read and benefits from strong community support and extensive documentation.

MATLAB, though powerful for numerical computation and data visualisation, it has a more complex Object Oriented Programming (OOP) scheme. It has less active and open community support compared to Python. This results in fewer readily available resources and support for ML development [47].

The development platforms to be used include:

a. **Visual Studio Code (VS Code):** This platform will be used for initial code development and data preprocessing. It allows for modular code design and supports robust extensions for Python development. Version control with Git will also be utilized to keep track of code versions.

b. **Google Colab Pro:** This platform will be employed for model training and testing. It provides access to high-performance GPUs and TPUs, which accelerate model training and reduce computational time. Its scalable computational resources can accommodate more complex models and larger datasets as needed.

## 3.3 Process Breakdown

This section provides a comprehensive overview of the key processes involved in the bird species classification system, detailing each subsystem's role and functionality. The classification system is built on a foundation of carefully structured components, from data preparation to performance evaluation.

### 3.3.1 Core Data Preparation and Preprocessing

This subsection outlines the essential steps involved in preparing the audio data for classification, ensuring that the raw recordings are transformed into a format suitable for machine learning models. Fig. 3.2 shows the approach to prepare both annotated and non-annotated audio data for bird species classification for feature extraction.

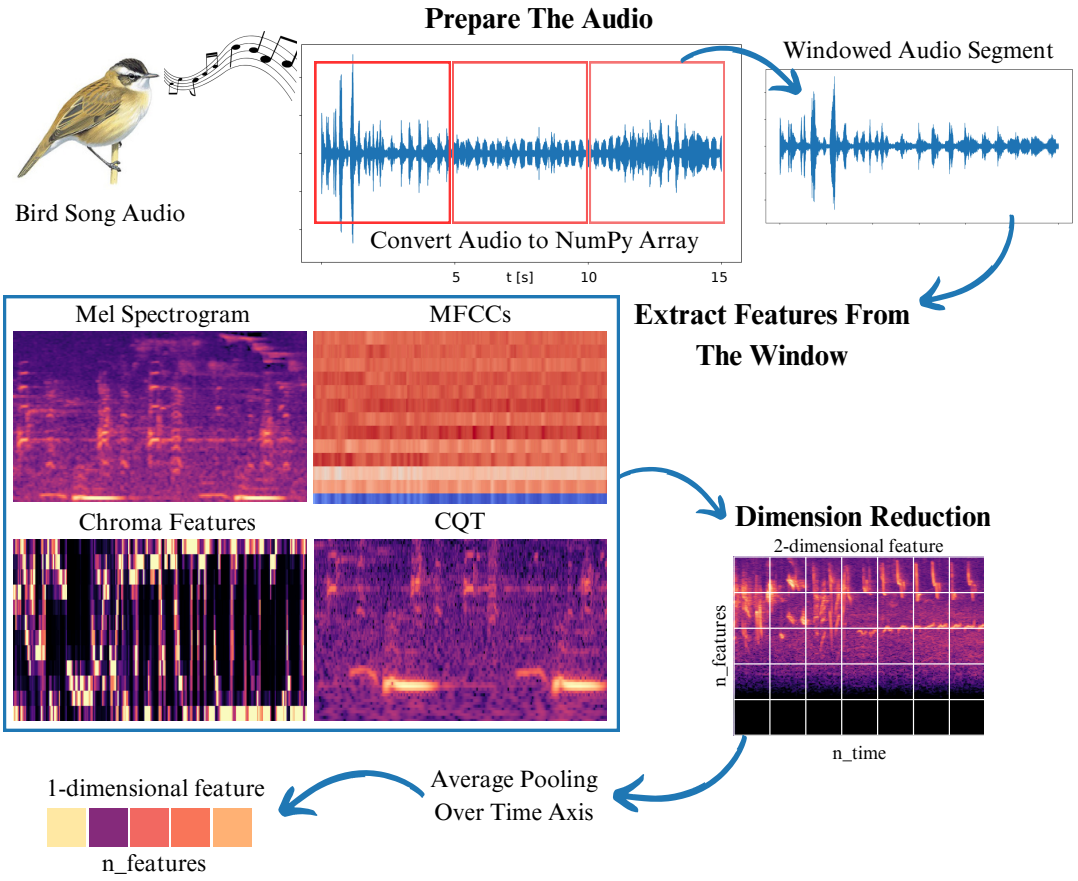



Figure 3.2: Preprocessing and feature extraction [48]

From Fig. 3.2, data is processed by segmenting entire audio files into fixed-size windows. Various window sizes are explored to determine the optimal length for classification. For annotated data, segments containing the bird calls are extracted based on the provided annotations. These extracted segments are then windowed to ensure consistent input sizes. If the audio segment is shorter than the window, it will be padded with pink noise to reach the window length.

Following segmentation, a range of features will be extracted from each windowed audio segment. These features include Mel spectrograms, MFCCs, chroma frequencies, and CQT.

To provide flexibility in model inputs and to investigate the impact of dimensionality on classification performance, two versions of each feature set are prepared. The first version retains the original 2D representation of each feature, preserving the full time-frequency information. The second version averages the 2D representations across the

time axis, resulting in a compact 1D feature set that may be more suitable for certain models or help to reduce computational complexity.

### 3.3.2 Initial Model Evaluation on Imbalanced Data

The dataset for this project presents a significant challenge due to the unequal representation of bird species in terms of sample size and duration. This imbalance necessitates a careful approach to model evaluation and selection.

### 3.3.3 Feature and Model Exploration

To address the dataset imbalance and identify the most effective approach, a systematic approach will be employed to evaluate the performance of various models by testing different combinations of window sizes, feature sets, and dimensionality. This process aims to identify which configurations yield the best results for bird species classification, and is as follows:

1. **Initial Model Training**: Multiple models will be trained on the imbalanced dataset using different combinations of features, window sizes, and dimensionality.
2. **Performance Tracking**: During the training process, the performance of each model will be closely monitored and recorded using comparable metrics.
3. **Configuration Ranking**: Based on the performance of the models trained, there will be internal ranking within the models to identify the top performers and the underperformers.
4. **Pruning Underperformers**: To optimise computational resources, the lowest-performing configurations will be eliminated, focusing effort on the most promising approaches.

The initial focus will be on traditional machine learning models, specifically starting with RF, XGBoost, SVM, and ending with KNN. From that, the general performance of all the feature combinations will be understood and that information can be used to focus the learning efforts of DL models on worthwhile combinations.

#### Hyperparameter Tuning

For all models, the best-performing combinations of features and window sizes will undergo hyperparameter tuning. This process aims to enhance model performance by optimising the selection of parameters, ensuring that each model is trained under the most effective conditions. It will involve techniques such as grid search to identify the optimal hyperparameter settings for each model.

This comprehensive approach to model training and evaluation is designed to rigorously assess the efficacy of various algorithms and configurations, ultimately identifying the most suitable methods for bird species classification.

#### Model Evaluation

The models that showed the best performance from the training and tuning phases will undergo rigorous evaluation with both annotated and unannotated data.

Testing on annotated data will serve as an indication of the model's performance when identifying a single bird call from a recording. Testing on unannotated data will simulate real world applications, where models are tested on uncontrolled excerpts that represent natural field conditions.

### 3.3.4 Model Refinement and Data Balancing

Following the evaluation of the models on the imbalanced data, the top two best-performing models will be retrained and evaluated on a balanced dataset to determine the impact of addressing class imbalance.

#### Data Balancing and Augmentation

The data will be balanced by oversampling the minority classes in the dataset, allowing to represent the underrepresented bird species. Using the new dataset, audio augmentation techniques will be implemented to increase the diversity and potentially improve the generalisation of the models. This will be done by performing time stretching, frequency pitching, and noise injection techniques.

**Comparative Analysis of Models**

After achieving a balanced training set, the focus will shift to model interpretability, which is essential for applications involving biological data. This will involve comparing the models to quantify the effect of data balancing and augmentation on model performance and generalisation capabilities.

### 3.3.5 Comprehensive Analysis and Summary

The final phase of the project involves a comprehensive analysis and conclusion drawing process. The performance of models trained on imbalanced data will be meticulously compared against those trained on balanced and augmented datasets. By evaluating these aspects, this study aims to draw robust conclusions about the most effective approaches for bird species identification across various scenarios, carefully weighing the trade-offs between model complexity, data preprocessing techniques, and overall performance.

To ensure reproducibility of the research, version control is implemented for both code and data management using Git and Good Drive respectively. Git allows one to track changes and maintain a clear history of code modifications, while Google Drive serves as a centralised location for consistent and accessible data storage. Random seed management is also incorporated to control the inherent randomness in model training and feature extraction.

## 3.4 Chapter Summary

The methodological framework and design choices presented in this chapter strike a careful balance between addressing the complexities of bird audio classification and employing cutting-edge tools and techniques. Key decisions were made to maximise the accuracy and robustness of the classification models, while accounting for practical project constraints.

With these foundational design principles established, the next phase transitions into the practical implementation of these approaches. This will involve applying the chosen models, refining them through evaluation, and optimising their performance.

# Chapter 4

# Project Implementation

Building on the comprehensive methodological framework outlined in the previous section, this implementation phase outlines the practical steps undertaken to execute the proposed approaches. It provides a detailed account of the tools, techniques, and workflows used to implement the research design. The diagram presented in Fig. 3.1 from the previous chapter will be used to guide this section.

## 4.1 Exploratory Data Analysis (EDA)

The first step in the implementation process was to perform a thorough EDA on the dataset to validate the data at hand. The WMWB dataset included a metadata file containing the list of audio files with species labels, as well as a separate text file with the species name, start time, and end time of each bird audio excerpt annotated within a single audio recording. To prepare the data for further processing, the information from the text file needs to be extracted and attached it to the respective audio recordings in the dataframe. This step ensured easy access and cross-referencing of the metadata throughout the implementation.

### 4.1.1 Data Preparation

To ensure consistency and computational efficiency, the following was implemented using the `librosa`, `pandas` and `numpy` libraries:

- **Downsampling and Channel Conversion**: All audio recordings were downsampled to a uniform sample rate of 22050 Hz and converted to mono, as the channels were not expected to contribute significantly to the model's ability to generalise.
- **Duplicate Check**: The duration of each audio sample was calculated and compared to ensure uniqueness, confirming that there are no duplicate samples.
- **Audio Conversion**: The audio files, originally in .mp3 format, will be converted to .npy files to facilitate efficient preprocessing. This is done by looping over each audio sample in the dataset and replacing the file extension accordingly to allow for efficient access.

### 4.1.2 Dataset Splitting

The dataset will be split using a 70/30 ratio, with 70% of the data used for training and validation and the remaining 30% reserved for testing. This ensures sufficient data for model evaluation while allowing adequate data for model training. Within the training data, a 80/20 intra-species split will be performed to ensure that all species are represented in both the training and validation sets. This approach ensures that the validation set is not skewed by an overrepresentation of certain species, thereby enabling better generalisation of the models across species. This split will be done using the `train_test_split` function from the `sklearn` library.

## 4.2 Feature Extraction Class

For ease of processing, a feature extraction class was created in Python. This class handles various tasks such as windowing the audio data, padding with pink noise, and extracting key audio features like MFCC, Mel-spectrogram, Chroma, and CQT. The class can be initialised with several parameters to customise the feature extraction process, including the window size, overlap, and sample rate. One can also choose whether to augment the data, normalise the features, or to perform dimension reduction via average pooling. The code for the class can be found in Appendix D. A brief explanation of the functions will be explained below:

- `normalize_audio`: This function normalises the audio samples to a [0, 1] range. This is particularly useful when padding the audio with noise to maintain a uniform scale across different samples.
- `generate_pink_noise` and `pad_with_noise`: Pink noise is generated by applying a filter to white noise. This is done using `np.fft` function to perform frequency scaling on the Fourier transform of the white noise. The filter shapes the noise to follow a 1/f frequency distribution. If an audio sample is shorter than the required window size, pink noise is used to pad it.
- `librosa.util.frame`: Used to window the audio, using the window size and overlap (hop length) as inputs.
- `extract_mfcc`: This function extracts the MFCCs from the windowed audio using `librosa`. This function generates 20 MFCCs per hop length. Hop length in this case is the number of samples between successive frames. A smaller hop length captures more temporal information. The hop length chosen for this project is 256 samples, which is half the default value.
- `extract_chroma`: This function uses `librosa` to produce 12 chroma bins per frame, which is the default value.
- `extract_cqt`: This function uses `librosa` to produce the default 84 bins per time frame. This covers 7 musical octaves and is sufficient to capture high resolution frequency content.
- `extract_melspectrogram`: This function uses `librosa` to produce the default 128 bins per time frame. This balances the high frequency resolution, with compact feature size.
- `avgpooling`: Average pooling is applied as an optional step for dimensionality reduction. This is done by calculating the mean across the time axis and pools the values along the time dimensions, effectively reducing the `n_time` dimension to a single value.
- `random_augmentation`: Applies random augmentation to the audio, such as pitch shift, time stretch, or noise addition.

The feature extraction class is designed to efficiently handle the extraction and storage of audio features and their corresponding species labels. Below is a summary of its key functionalities and how it organises the extracted data:

- **Data Storage Structure**: Features are stored in a dictionary with keys for each specific feature and values as lists of computed data for each audio window. Species labels are then kept in a separate array for proper alignment with features, allowing efficient retrieval for model input and training.
- **Feature Output Options**: The class returns both average pooled and non-average pooled data. It checks for average pooling during extraction; if enabled, dimension reduction is applied. Outputs include the original non-average pooled features and the average pooled features. This ensures the extraction process is run once, providing both data versions for further analysis and model training.

For this study, two window sizes were selected: 3 seconds and 1 second, each with a 50% overlap. While larger window sizes were initially considered to capture more information, they would be unsuitable due to the variability in bird call durations. Larger windows would risk overwhelming short bird calls with background noise, as the longer window would contain more non-signal portions, reducing the effectiveness of the classification.

## 4.3 Label Encoding and Feature Storage

Upon extracting the features, label encoding was implemented. This is the process of converting categorical labels (the species names) into a numerical format so that they can be understood by ML algorithms, as they often cannot directly interpret string values. Once the labels are encoded, they can be easily integrated into the training and validation datasets. This was done using a `LabelEncoder` instance from `sklearn`. It is fitted to the training labels and then learns the mapping of each unique species to a corresponding integer (see Tab. C.1)

Two dictionaries, `merged_dict_1D` and `merged_dict_2D`, are created to combine the training and validation sets of average pooled and non-average pooled features, respectively. The `pickle` library is then used to serialise and save the merged dictionaries to .pkl files, enabling efficient storage and retrieval of the datasets for future use.

## 4.4 Initial Model Evaluation on Imbalanced Data

To gain an understanding of how different models perform on the bird species classification task, various feature combinations were explored to evaluated model performance using a range of metrics. The table below shows all the feature combinations that will be explored for both annotated and unannotated data during the initial evaluation. Each combination represents different sets of features used for training and validation.

| | |
|---|---|
| chroma | mfcc_melspectrogram |
| cqt | melspectrogram |
| chroma_cqt | melspectrogram_cqt |
| mfcc | melspectrogram_chroma_mfcc |
| mfcc_chroma | melspectrogram_cqt_mfcc |
| melspectrogram_chroma | all_features |

Table 4.1: Feature Combinations

### 4.4.1 Evaluation Metrics

To evaluate the model's performance on the classification task as a whole, accuracy will be the primary metric. To ensure accurate classification of individual species, the model's performance will be evaluated using the F1-score, precision, and recall metrics.

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It reflects how often the model correctly identifies a species when predicting that species.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall, also known as sensitivity, measures the proportion of actual positive cases correctly identified by the model. It indicates the model's ability to detect all relevant instances of a species.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

The F1-score is the harmonic mean of precision and recall, balancing both false positives and false negatives. This metric is particularly useful when dealing with imbalanced

datasets, as it provides a more balanced view of the model's performance.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To maintain consistency across model evaluations, an `evaluate_model` function was implemented. This function accepts the true labels, predicted probabilities, and predicted labels as inputs and returns a comprehensive classification report, including key evaluation metrics such as the F1-score and AUC.

Listing 4.1: Function to evaluate the model based on AUC and F1-score

```
def evaluate_model(val_y, val_yhat, val_yhat_result, num_classes=20):
    print(classification_report(val_y, val_yhat_result))

    # Calculate AUC for multiclass classification
    auc_score = roc_auc_score(val_y, val_yhat, average='weighted')

    # Calculate F1-score with 'weighted' average for imbalanced dataset
    f1 = f1_score(val_y, val_yhat_result, average='weighted')
    val_score = {'f1': f1, 'auc': auc_score}
    return val_score
```

The function uses `sklearn.metrics.classification_report` to calculate the precision, recall and F1-score of each class. It also uses `sklearn.metrics.roc_auc_score` to calculate the AUC to help understand how well the model distinguishes between classes, and ensures that the score accounts for class imbalance by weighting each class's contribution based on its size. This allows for a standardised process, and consistent comparison between models.

### 4.4.2 Random Forest: Establishing the Baseline

To establish a baseline, a RF model was selected and the model was implemented using the `sklearn` library. The initial parameters were set as follows:

- `n_estimators=100`: The number of trees in the forest, chosen to balance computation time and performance.
- `max_depth=None`: Allowing the trees to grow fully to capture complex interactions in the data.

- `criterion='entropy'`: This criterion was selected to evaluate the quality of splits in the trees, as it considers the information gain, which can be effective in capturing the underlying patterns in the data.

### 4.4.3 XGBoost: Further Understanding and Fine-Tuning

Following the RF model, XGBoost was employed to further investigate how different feature combinations impact model performance. The model was implemented using the `xgboost` library and the initial parameters used were:

- `n_estimators=100`: A moderate number of boosting rounds to ensure thorough learning.
- `booster='gbtree'`: This specified the use of the tree-based booster, optimising the model for structured data.

### 4.4.4 SVM: Focused Evaluation

After eliminating feature combinations that consistently underperformed in the previous models, a SVM was trained. The model was implemented using the `sklearn` library and the initial parameters used were:

- `C=10`: This regularisation parameter allows the SVM to prioritise minimising classification errors
- `kernel='rbf'`: The radial basis function kernel was selected due to its effectiveness in non-linear classification tasks.

### 4.4.5 KNN: Final Traditional Model Evaluation

Lastly, a KNN model was used to conclude the traditional model evaluation process. The model was implemented using the `sklearn` library and the initial parameters used were:

- `n_neighbors=15`: This was chosen to balance capturing sufficient neighbors for robust classification and avoiding overfitting.

- `weights='distance'`: This setting ensures that closer neighbors have more influence on the classification.
- `algorithm='auto'`: To allow the model to automatically select the most appropriate algorithm based on the input data.

After evaluating the performance of various feature combinations in the traditional models, the majority of underperforming combinations will be eliminated. Only the best-performing combinations will be retained for training deep learning models.

### 4.4.6 Feedforward Neural Network (FFNN) Model

This will serve as a baseline for evaluating how neural networks handle the classification task compared to traditional models. The architecture of the initial model, depicted in Fig. 4.1, is designed with key parameters chosen to enhance training stability and accuracy.

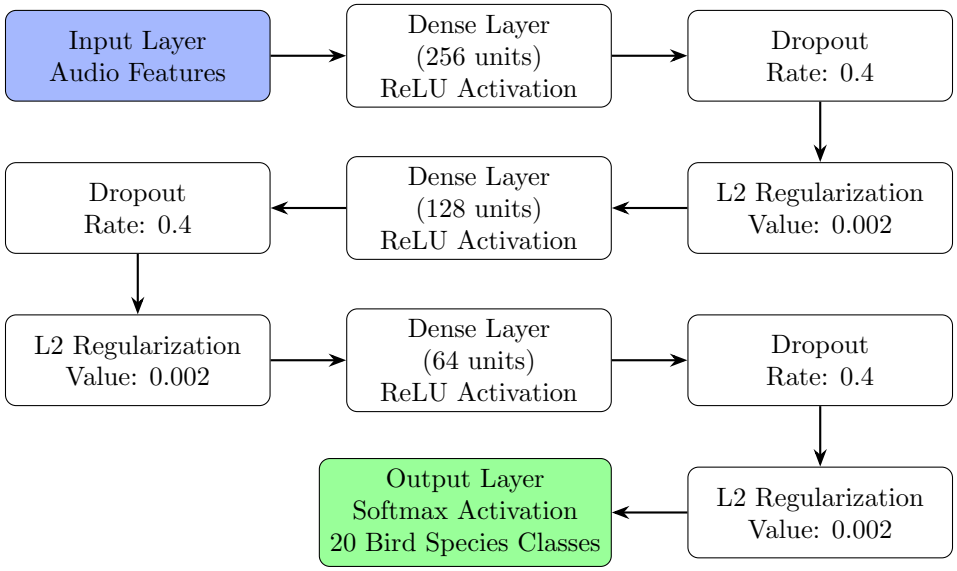



Figure 4.1: FFNN architecture

Implemented using the `keras` framework, the model utilises the Sparse Categorical Cross-Entropy loss function, which efficiently handles multi-class classification tasks with integer-encoded labels. To maintain consistency in evaluation metrics, this loss function is adopted as the standard across all DL models in this study.

Training is optimised using the Adam algorithm, chosen for its ability to dynamically

adjust learning rates through adaptive moment estimation. The learning rate is deliberately set to 0.0001, enabling the model to make measured weight updates that preserve training stability while maintaining sufficient plasticity to capture underlying patterns in the data.

### 4.4.7 Convolutional Neural Network (CNN) Model

Following the FFNN, a CNN will be implemented. CNNs are particularly effective for feature extraction from structured data, making them a highly suitable choice for this classification task. The architecture of the initial CNN model is illustrated in Fig. 4.2.

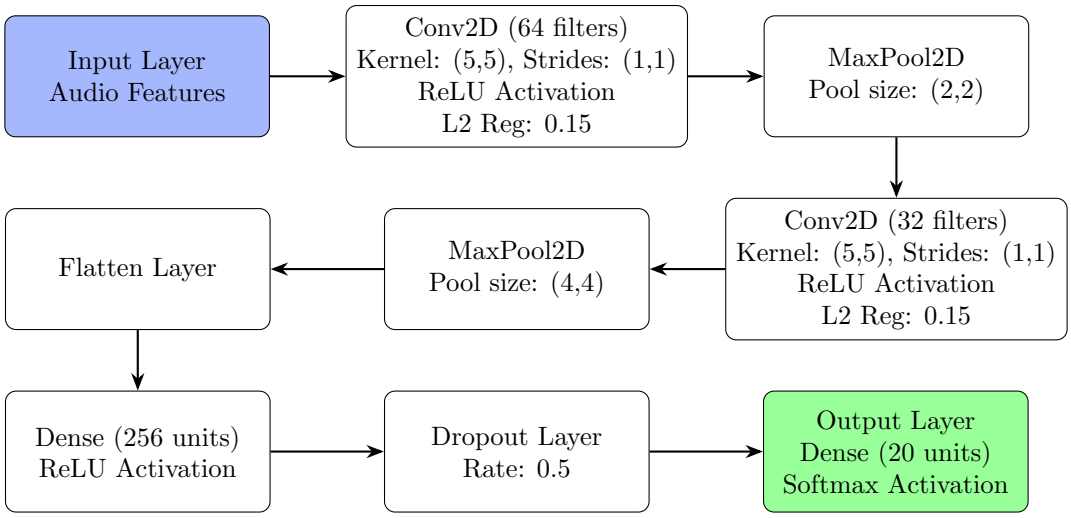



Figure 4.2: CNN architecture

Training is optimised using the Adam optimiser with a learning rate of 0.00005. This choice of learning rate is designed to facilitate gradual convergence and enhance the model's ability to learn intricate patterns in the data while minimising the risk of overshooting optimal solutions.

### 4.4.8 Transfer Learning Models

Lastly, Transfer Learning will be employed, leveraging VGG-16 and ResNet-50. Fine-tuning these models on the bird species classification task allows for the use of pre-learned features from large-scale datasets like ImageNet, which is especially beneficial when working with smaller datasets.

ResNet-50 employs bottleneck convolutional layers, each composed of three convolutional layers, with batch normalisation and ReLU activation applied after each layer. The "50" in ResNet-50 refers to the fact that the network includes 50 such layers, structured into bottleneck blocks stacked sequentially. This architecture allows for efficient feature extraction while mitigating the vanishing gradient problem through residual connections.

The initial model architecture for this project is depicted in Fig. 4.3, using the same compilation parameters as the CNN.

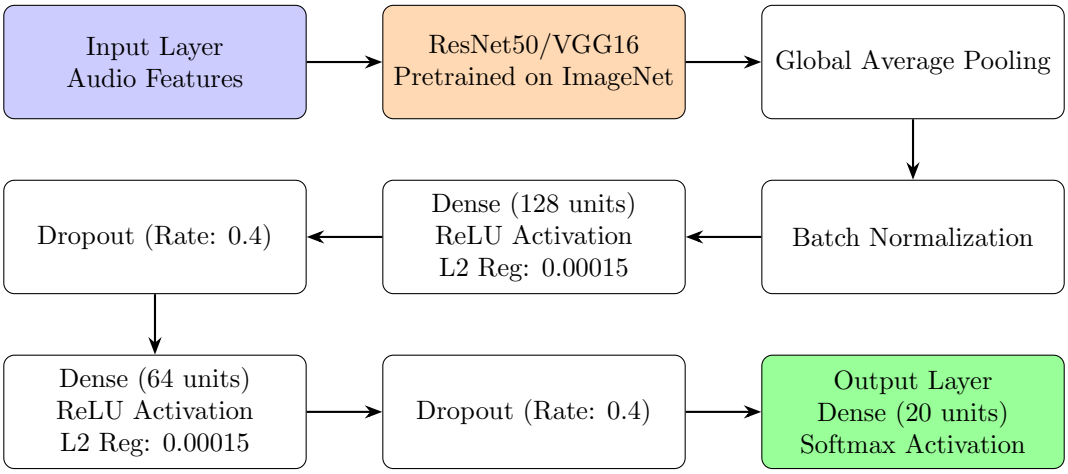



Figure 4.3: Transfer learning-based audio classification model architecture

## 4.5 Model Evaluation on Balanced/Augmented Data

After training the top-performing models from each architecture and algorithm, testing will be conducted using both annotated and unannotated data, as outlined in Chapter 3. Once model performance has been evaluated on the test set, the focus will shift to enhancing robustness by examining the effects of a more balanced dataset and the application of audio data augmentation techniques. This will help refine the models' generalisation capabilities and improve their performance on imbalanced data.

### 4.5.1 Balancing and Enhancing Training Data

The balancing process is done by oversampling the minority classes to reduce class imbalance. Using the number of sample for the majority class as a reference, the target count for oversampling is set to 70% of that value. This is to avoid perfect balance, which

could simulate more realistic conditions while improving class distribution. This is done using the `resample()` function from `sklearn`.

The training data is enhanced by augmentation using `random_augmentation` described in the feature extraction class. The function randomly selects one of five possible augmentations: pitch shift up, pitch shift down, adding noise, stretching, or shrinking. The function is shown below:

Listing 4.2: Function to randomly augmentent an audio window

```
def random_augmentation(self, audio):
    aug_type = np.random.choice(['pitch_up', 'pitch_down', 'noise',
                                 'stretch', 'shrink'])
    if aug_type == 'pitch_up':
     pitch_shift = np.random.uniform(1, 4) # Between 1 and 4 semintones
     audio = librosa.effects.pitch_shift(audio, n_steps=pitch_shift)
    elif aug_type == 'pitch_down':
     pitch_shift = np.random.uniform(-4, -1) # Between 1 and 4 semintones
     audio = librosa.effects.pitch_shift(audio, n_steps=pitch_shift)
    elif aug_type == 'noise':
     noise = self.generate_pink_noise(len(audio)) # Add random noise
     audio = audio + 0.01 * np.random.uniform(0.2, 0.5) * noise
    elif aug_type == 'stretch':
     stretch_factor = np.random.uniform(1.1, 1.5) # Randomly stretch audio
     audio = librosa.effects.time_stretch(audio, rate=stretch_factor)
    elif aug_type == 'shrink':
     stretch_factor = np.random.uniform(0.8, 0.9) # Randomly shrink audio
     audio = librosa.effects.time_stretch(audio, rate=stretch_factor)
    return audio
```

The function returns the augmented audio sample after applying one of the random transformations This randomisation ensures diverse augmentations are applied to the data, providing varied training examples for the model.

### 4.5.2 Final Model Evaluation on Balanced and Augmented Data

The final phase of this investigation focuses on evaluating model performance using the standardised testing set, with particular emphasis on models trained on balanced and augmented data. This evaluation aims to quantify the effectiveness of these preprocessing techniques on model performance and generalisation capabilities.

The assessment methodology maintains consistency by employing the previously established evaluation metrics to models trained on data that has undergone minority class oversampling and strategic augmentation techniques.

## 4.6 Chapter Summary

The implementation chapter presents a comprehensive overview of the bird species classification system development process. Through detailed documentation of each development phase - from initial data preparation to final model evaluation - this chapter provides insights into both the technical framework and the methodological decisions that shaped the research outcome. A systematic approach to implementation, anchored by modular components such as the dedicated feature extraction pipeline, not only streamlined the development process but also enhanced the system's maintainability. This structured documentation serves to ensure both the transparency of the methods and the reproducibility of the findings.

# Chapter 5

# Project Results

This chapter presents the results of the various models trained and evaluated for bird species classification. The performance of both traditional machine learning models and DL models will be explored across different data configurations. A detailed discussion of these results follows, emphasising model accuracy, overfitting, and the challenges posed by data imbalance. The complete results of all the models can be found in the Appendix.

## 5.1 Experimental Setup

The experimental phase of this research focuses on evaluating the performance of various machine learning models. Establishing a clear experimental setup is crucial as it ensures the reliability and reproducibility of the results while providing essential context for interpreting model performance. Moreover, understanding the computational environment helps contextualize training times, resource constraints, and any potential limitations that might influence model selection and optimisation strategies.

The development environment consisted of two main platforms, each serving different phases of the project. Initial feature extraction was performed on a local workstation with the following specifications:

- **Processor**: AMD Ryzen 5 5600H with Radeon Graphics (3.30 GHz)
- **RAM**: 20GB (19.3GB usable)
- **System**: 64-bit operating system, x64-based processor

Due to the memory-intensive nature of model training and the need for GPU acceleration, Google Colab Pro was employed as the primary training environment. All models were trained using Colab's NVIDIA T4 GPU, which offers an optimal balance between computational power and resource efficiency.

## 5.2 Dataset Analysis and Characteristics

This section presents a detailed examination of the dataset composition, distribution patterns, and the implications of data partitioning for model development.

### 5.2.1 Dataset Distribution and Imbalance

An analysis of the dataset revealed that it contained 20 unique bird species, with significant variations in both sample count and audio duration. The *Himantopus himantopus* species represented the majority class with 70 samples, while the *Anas strepera* species formed the minority class with only 9 samples. In total, the dataset comprised 879 audio recordings, with durations ranging from 1.44 seconds to 1126 seconds (see Tab. C.1). The temporal characteristics of the recordings showed notable disparities. The shortest duration total duration per species was 537 seconds from the *Anas strepera* species. The longest duration was from *Acrocephalus arundinaceus* with 5200 seconds of audio (see Fig. C.1).

These variations in recording duration and sample count present two distinct types of imbalance:

- **Sample Count Imbalance**: The ratio between the most and least represented species is approximately 7.8:1
- **Duration Imbalance**: The ratio between the longest and shortest total duration is approximately 9.7:1

### 5.2.2 Data Splitting Results

The dataset was strategically partitioned into training and validation sets while maintaining class representation. The resulting distribution showed significant variations in available audio duration:

| Set | Largest Duration | Smallest Duration |
|---|---|---|
| Training | *Acrocephalus arundinaceus* (48 min) | Anas platyrhynchos (3 min) |
| Validation | *Coracias garrulus* (13 min) | Several species limited to 1 minutes |

Table 5.1: Training and validation split durations

The total audio duration available for training is 351 minutes and 99 minutes for validation. The splitting strategy maintained a consistent ratio across classes while ensuring that there is no overlap between the sets and the natural variation within each class is preserved.

## 5.3 Feature Extraction Analysis

The feature extraction process plays a crucial role in transforming raw audio data into meaningful representations for machine learning models. The feature extraction process, depicted in Fig. 3.2, generated multiple audio representations with the following dimensions for each window size:

| Feature | Original [3 s, 1 s] | Pooled | Flattened [3 s, 1 s] |
|---|---|---|---|
| Mel-Spectrogram | [(128, 259), (128, 87)] | (128,) | [(33152,), (11136)] |
| MFCC | [(20, 259), (20, 87)] | (20,) | [(5180,), (1740)] |
| CQT | [(84, 259), (84, 87)] | (84,) | [(21756,), (7308)] |
| Chroma | [(12, 259), (12, 87)] | (12,) | [(3108,), (1044)] |

Table 5.2: Feature dimensions

Understanding the feature dimensionality is essential for optimising machine learning models and managing computational resources efficiently.

## 5.4 Initial Model Evaluation on Imbalanced Data

This section outlines the performance of models trained on the imbalanced dataset. Results for traditional machine learning models and DL models are provided, with a focus on training accuracy, validation accuracy, and class evaluation metrics.

### 5.4.1 Baseline Performance (RF)

The RF model was chosen as a baseline to evaluate how well traditional machine learning algorithms perform on the imbalanced bird species classification task. During training, the model achieved a perfect accuracy of 100%, indicating that it could fully capture the patterns in the training data. Fig. 5.1 shows a heatmap of validation accuracy for each configuration. This visualisation provides a comprehensive overview of how the model performs across varying input feature sets and preprocessing strategies.

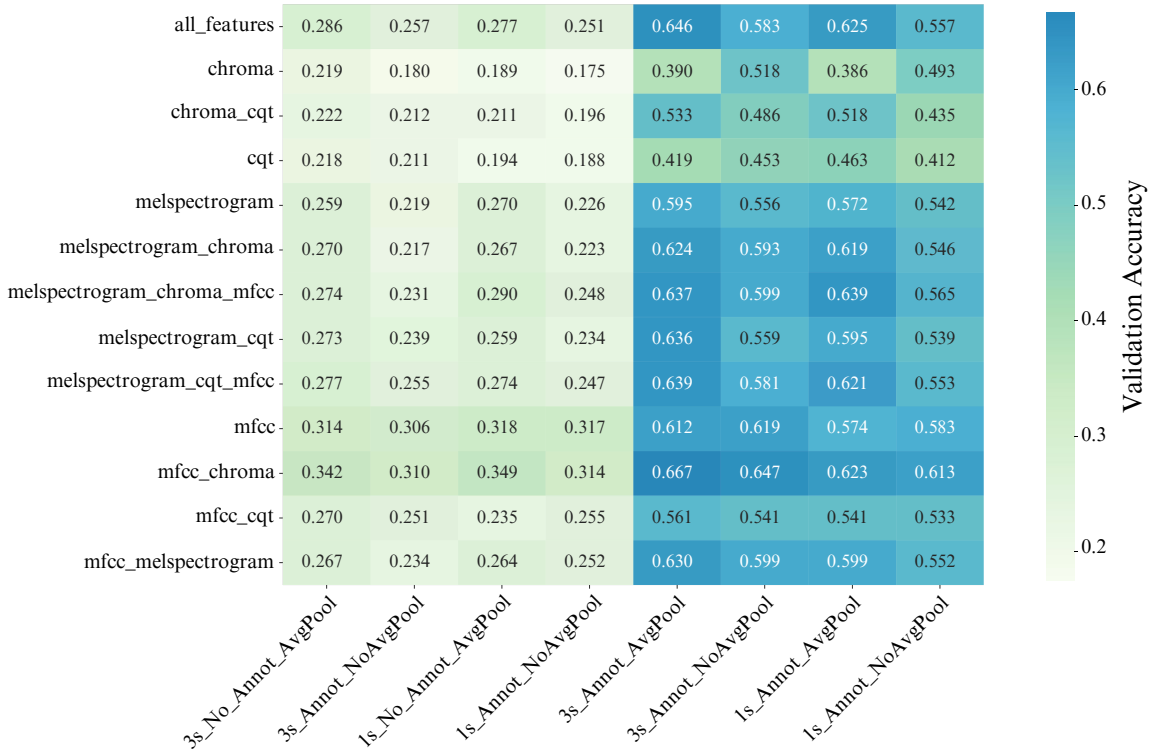



Figure 5.1: RF validation accuracy heatmap across all configurations

The heatmap reveals a disparity between the validation accuracies of unannotated and annotated data. The unannotated data achieved a maximum validation accuracy of 34.9%. In contrast, the annotated data attained a significantly higher validation accuracy of 66.7%, achieved with a 1D, 3-second window, and the MFCC and Chroma feature combination.

The original 2D data consistently underperformed compared to the average-pooled data. The highest validation accuracy obtained using the original 2D data was 64.7%,

while the lowest was 17.5%, marking the poorest performance in the entire training process. Chroma emerged as the worst-performing feature across all configurations.

The best-performing model achieved a perfect 100% precision for *Botaurus stellaris*, while *Anas strepera*, which had the fewest samples, scored 0% across all metrics. *Alcedo atthis* had a perfect recall of 100%, successfully identifying all instances of the species, though its precision was 82%, indicating strong but not flawless classification. The overall weighted F1-score was 0.66, reflecting moderate performance across most species. The AUC score of 94.9% indicates a strong ability to distinguish between classes.

Hyperparameter tuning was applied to the best-performing model (see Tab. D.1), resulting in a slight improvement in performance. Despite these adjustments, the model continued to show signs of overfitting, as indicated by a perfect 100% training accuracy. The validation accuracy increased to 67% when using the log loss criterion and increasing the number of estimators from 100 to 200. Despite the slight improvement, the macro F1-score for the model was 0.61, indicating challenges in accurately classifying certain species, particularly.

### 5.4.2 XGBoost Results on Imbalanced Data

The XGBoost classifier was implemented to further explore performance improvements.

(a) Best performing XGBoost models

| Window Size | Annotation | Feature Combo | Val Accuracy |
|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `all features` | 0.368 |
| 1 Second, 1D | No Annotation | `mfcc_chroma` | 0.334 |
| 3 Seconds, 2D | Annotation | `mfcc_chroma` | 0.712 |
| 1 Second, 2D | Annotation | `mfcc_chroma` | 0.696 |

(b) Worst performing XGBoost models

| Window Size | Annotation | Feature Combo | Val Accuracy |
|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `chroma` | 0.208 |
| 1 Second, 1D | No Annotation | `chroma` | 0.181 |
| 3 Seconds, 1D | Annotation | `chroma` | 0.389 |
| 1 Second, 1D | Annotation | `chroma` | 0.401 |

Table 5.3: Summary of validation accuracies for XGBoost models by window size

During training, and like the RF, the model achieved a perfect accuracy. In Tab. 5.3a,

the highest validation accuracy of 71.2% was achieved with a 2D annotated 3-second window with MFCC and chroma. From Tab. 5.3b, the `chroma` feature consistently resulted in the lowest validation accuracies. The lowest score of 18.1% was achieved with a 1-second window and no annotations. Even when annotations were applied, the chroma feature's performance remained subpar, with validation accuracies of 38.9% and 40.1% for 3-second and 1-second windows, respectively. In contrast to the RF, the average pooled data exhibited inferior performance compared to the original data.

The best model achieved its highest precision of 97% for *Coracias garrulus* and an impressive recall of 99% for *Alcedo atthis*, demonstrating its effectiveness in detecting these species. In contrast, *Anas strepera* exhibited the lowest performance, with precision, recall, and F1-score values of 17%, 7%, and 10%, respectively, reflecting challenges in identifying this species. The weighted F1-score of 0.71 and a macro average F1-score of 0.68, indicate a moderate performance across most classes and an improved model from the RF. The AUC score of 96% highlights a strong ability to distinguish between classes.

Hyperparameter tuning was performed on the XGBoost model (see Tab. D.2), leading to varying performance outcomes across different configurations. The model achieved perfect training accuracy of 100% across all tested estimators and boosters, indicating overfitting concerns. The best validation accuracy was recorded at 68.8% when using 100 estimators with the gbtree booster and no specified learning rate. The tuning process also indicated that the model continues to face difficulties with certain classes, underscoring the ongoing challenges in achieving accurate classification of specific species.

### 5.4.3 SVM Results on Imbalanced Data

Following the XGBoost classifier, the SVM model was implemented to further evaluate its performance in classifying the bird species. The key results from each window size are presented in Tab. 5.4a and 5.4b.

The best-performing SVM models indicate that configurations using the Mel-spectrogram, chroma, and MFCC features achieved commendable training accuracies, with the highest validation accuracy of 72.4% recorded for the 3-second window with annotations. In contrast, configurations without annotations consistently underperformed, with the highest validation accuracy reaching 34.7% for the 3-second window. The model did not achieve perfect training accuracy, indicating less overfitting than the previous models.

(a) Best performing SVM models

| Window Size | Annotation | Feature Combo | Train Acc | Val Acc |
|---|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `melspectrogram_chroma_mfcc` | 0.760 | 0.347 |
| 1 Second, 1D | No Annotation | `melspectrogram_chroma_mfcc` | 0.774 | 0.310 |
| 3 Seconds, 2D | Annotation | `melspectrogram_chroma_mfcc` | 0.958 | 0.724 |
| 1 Second, 2D | Annotation | `melspectrogram_chroma` | 0.976 | 0.722 |

(b) Worst performing SVM models

| Window Size | Annotation | Feature Combo | Train Acc | Val Acc |
|---|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `cqt` | 0.966 | 0.203 |
| 1 Second, 1D | No Annotation | `chroma` | 0.358 | 0.181 |
| 3 Seconds, 1D | Annotation | `chroma` | 0.605 | 0.396 |
| 1 Second, 1D | Annotation | `chroma` | 0.591 | 0.401 |

Table 5.4: Summary of accuracies for SVM models by window size

The worst-performing models reveal substantial challenges, particularly with limited features. For example, the 1D, 3-second window with no annotations and the chroma feature yielded a validation accuracy of 18.1%, which may suggest the model struggled to learn with the 12 features.

For hyperparameter tuning, cross-validation was implemented with $CV = 5$ to enhance the performance of the model. The tuning involved a parameter grid focusing on various values for the regularisation parameter $C$, and the kernel coefficient $\gamma$ (see Algorithm 1, and appendix A.3).

**Algorithm 1** Parameter Grid Definition

1: $C\_values \leftarrow [0.5, 4, 6, 10, 100, 150, 200, 300]$
2: $gamma\_values \leftarrow$ [`scale`, 1, 0.1, 0.01, 0.001, 1.5, 2, 2.5, 3]
3: $kernel\_values \leftarrow$ [`rbf`]
4: $param\_grid \leftarrow$ `empty dictionary`
5: Add to $param\_grid$: `'C'`: $C\_values$
6: Add to $param\_grid$: `'gamma'`: $gamma\_values$
7: Add to $param\_grid$: `'kernel'`: $kernel\_values$
8: **return** $param\_grid$ =0

The results from this tuning process indicated that the optimal hyperparameters were $C = 150$, $\gamma = 1$, and the RBF kernel, achieving a training accuracy of 100% and a validation accuracy of 64.07%. While the model demonstrated excellent training accuracy, the validation results suggest room for improvement in generalisation to unseen data.

### 5.4.4 KNN Results on Imbalanced Data

Based on the performance of earlier models, it was evident that the 2D unannotated data consistently underperformed and required excessive computation time without yielding promising results. Consequently, these configurations were excluded from further evaluation. A summary of the validation results for the model is provided below:

(a) Best performing KNN models

| Window Size | Annotation | Best Feature Combo | Val Accuracy |
|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `mfcc_chroma` | 0.313 |
| 1 Second, 1D | No Annotation | `mfcc_chroma` | 0.322 |
| 3 Seconds, 1D | Annotation | `melspectrogram_chroma_mfcc` | 0.596 |
| 1 Second, 2D | Annotation | `mfcc_chroma` | 0.637 |

(b) Worst performing KNN models

| Window Size | Annotation | Worst Feature Combo | Val Accuracy |
|---|---|---|---|
| 3 Seconds, 1D | No Annotation | `cqt` | 0.184 |
| 1 Second, 1D | No Annotation | `chroma` | 0.184 |
| 3 Seconds, 1D | Annotation | `chroma` | 0.361 |
| 1 Second, 1D | Annotation | `chroma` | 0.360 |

Table 5.5: Summary of validation accuracies for KNN models

The feature combination of MFCC and chroma achieved the highest validation accuracy at 63.7%. Meanwhile, feature sets relying solely on chroma or CQT underperformed across the board, with validation accuracies frequently below 50%. Additionally, the models utilising average pooling tended to slightly outperform those without it.

The best-performing model showed an AUC score of 89% and an F1-score of 0.61, showing good balance between sensitivity and specificity. Multiple species had perfect precision, and the lowest precision obtained by *Acrocephalus melanopogon* with 39%.

The optimal number of neighbours was found to be 1, after considering various values between 1 and 40. With 1 neighbour, the model achieved a training accuracy of 100% and a validation accuracy of 60.5%. When using SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance, the model achieved a validation accuracy of 59.6%, showing improvement in handling imbalanced data but still slightly underperforming.

### 5.4.5 FFNN Results on Imbalanced Data

In the shift towards implementing a FFNN, only a select set of feature combinations was retained based on prior performance. Combinations that consistently underperformed were discarded, while the top-performing feature sets were carried forward. These included:

- All features together
- Mel spectrogram alone
- Mel spectrogram with chroma
- Mel spectrogram, chroma, and MFCC combined
- Mel spectrogram, CQT, and MFCC combined

The models were trained for 100 epochs using a batch size of 32 and the results are shown in Fig. 5.2. Each epoch took an average of 1 second to finish computing. The highest validation accuracy of 74.1% was achieved with the annotated, original, `melspectrogram_chroma_mfcc` feature combination. Annotated data consistently yielded validation accuracies above 70%, with validation accuracies remaining below 41% for unannotated combinations.

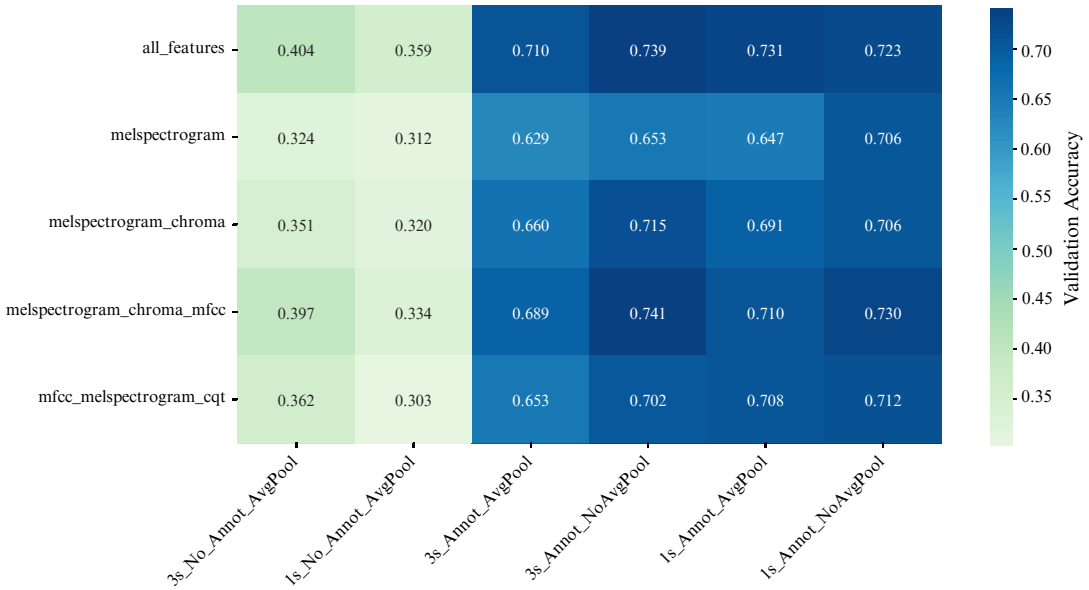



Figure 5.2: FFNN validation accuracy heatmap across retained configurations

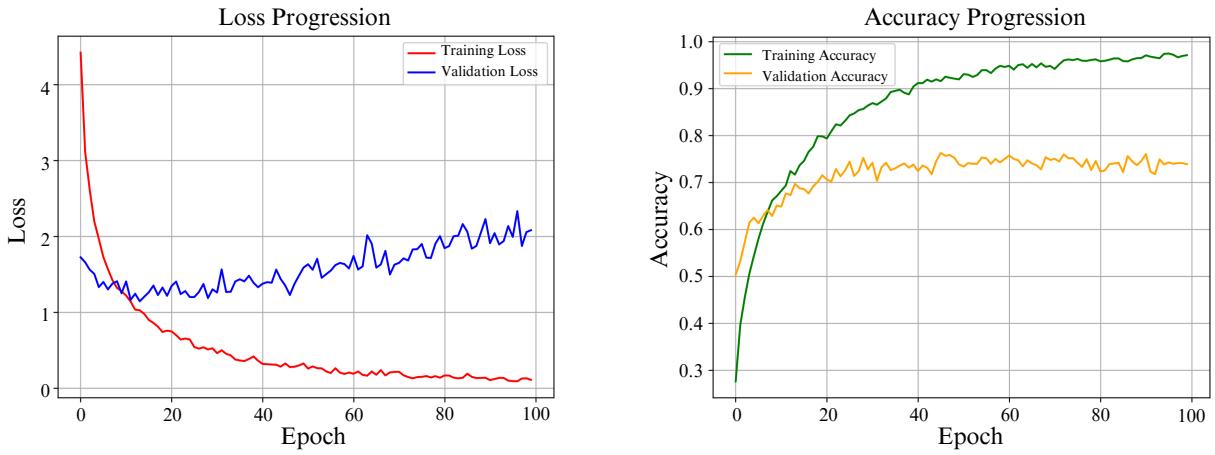



Figure 5.3: Best initial model training and validation history for 100 epochs

The classification report for the best-performing FFNN model highlights a weighted F1-score of 0.72, indicating balanced performance across most classes. Certain species, such as *Alcedo atthis*, *Botaurus stellaris*, and *Ixobrychus minutus*, exhibited excellent performance, with precision, recall, and F1-scores above 0.90. Notably, *Botaurus stellaris* achieved an F1-score of 0.98, showing near-perfect classification. Some classes, particularly class 5 and class 6, showed significant underperformance, with *Anas strepera* having an F1-score of 0.00 due to low recall and precision. The model achieved an AUC score of 95.6%, indicating strong overall discriminative ability between the classes.

Hyperparameter tuning was conducted on the FFNN, varying parameters such as hidden layer size, number of epochs, learning rate, L2 regularisation, and dropout rates (see Tab. D.3). The best performing model achieved a validation accuracy of 74.4%, which is a slight increase in accuracy compared to the initial model.

### 5.4.6 CNN Results on Imbalanced Data

The CNN was trained using a batch size of 32 over 100 epochs, with an average of 7 seconds per epoch. Unannotated data was excluded from the training process after the FFNN exhibited significant model loss when this data was included. The average pooled data was also excluded in favour of training with a 2D CNN. Fig. 5.4 shows the model loss and accuracy during training, showing good convergence compared to the FFNN. The table below summarises the CNN models' performance on the annotated data.

| Features | 3 Second Window | 1 Second Window |
|---|---|---|
| allfeatures | 0.715 | 0.732 |
| melspectrogram | 0.747 | 0.715 |
| melspectrogram_chroma | 0.724 | 0.718 |
| melspectrogram_chroma_mfcc | 0.679 | 0.705 |
| melspectrogram_cqt_mfcc | 0.756 | 0.729 |

Table 5.6: CNN summary of validation accuracies

Extensive hyperparameter tuning was performed to further optimize the CNN's architecture and parameters. This involved varying convolutional layers, dense units, learning rates, regularisation techniques, and dropout rates (see Tab. D.4). The highest validation accuracies recording from tuning still did not exceed the initial training. Lower learning rates generally led to more stable training, with the learning rate of 0.0001 producing the highest validation accuracy of 73.8% after 90 epochs. Some models, such as the one with a learning rate of 0.00005, showed perfect training accuracy but lower validation accuracy 68.1%, indicating overfitting to the training data. More complex CNN architectures with additional layers did not consistently yield higher validation accuracy, with a third convolutional layer slightly reducing performance in some cases.

From the classification report of the best performing model, the overall AUC score of 0.965 further confirmed the model's robust discriminatory power. Over 50% of the species obtained an F1-score greater than 0.8, and the lowest recorded score was 50% by *Ardea purpurea*.

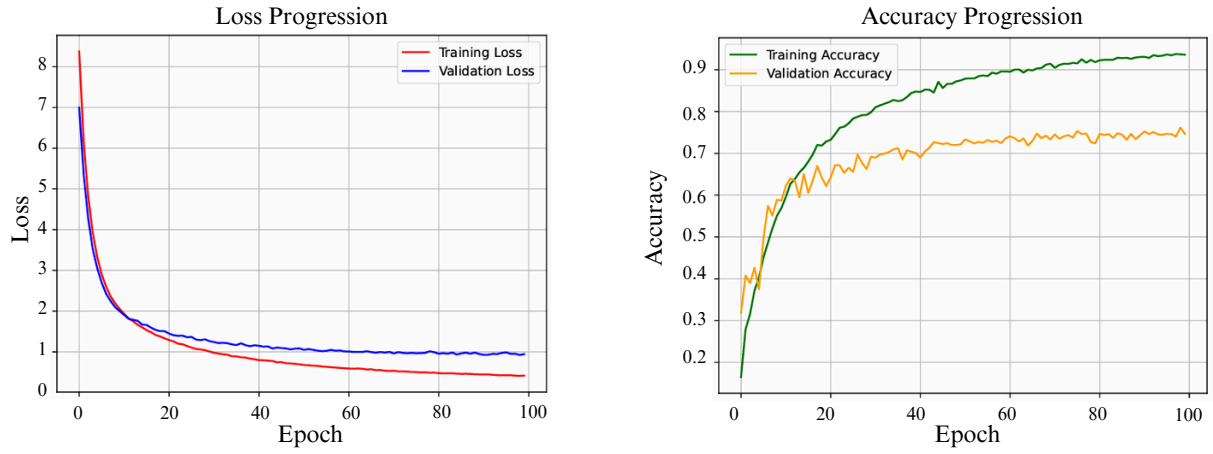



Figure 5.4: Best performing CNN model training and validation history for 100 epochs

### 5.4.7 Transfer Learning Results on Imbalanced Data

Similar to the previous DL models, these models were training with a batch size of 32. Since the models are pre-trained, fewer epochs were required for convergence, so 60 epochs were selected for training, with an average training time of 52 seconds per epoch. Additionally, as the models were trained on 3-channel images, only the best-performing three-feature combinations were used for training. Both architectures reached perfect training accuracy. The validation results are presented below:

Table 5.7: VGG-16 and ResNet-50 Summary of Validation Accuracies

| Features | VGG-16 | ResNet-50 |
|---|---|---|
| melspectrogram_chroma_mfcc | 0.847 | 0.819 |
| melspectrogram_cqt_mfcc | 0.873 | 0.834 |

Due to the superior performance of the 3-second annotated window size, it was exclusively used for training these models. All models achieved validation accuracies exceeding 80%, surpassing the performance of previously trained models. Notably, the VGG-16 architecture, using the Mel-spectrogram, CQT, and MFCC features, achieved the highest validation accuracy of 87.3%. This was the best performing model, even after tuning.

The classification report for the best-performing model revealed that 70% of the species achieved an F1-score above 0.8, with several species attaining perfect precision and recall. The lowest F1-score was recorded for *Anas platyrhynchos* at 0.48. Remarkably, *Anas strepera*, which had consistently struggled in previous models with F1-scores below 0.5, achieved a significant improvement, reaching an F1-score of 0.93, despite its limited representation in the dataset. This highlights the model's enhanced ability to classify even underrepresented species accurately. The model achieved a strong a weighted F1-score of 0.87 and an impressive AUC score of 0.987. The ResNet models demonstrated performance on par with the VGG-16, delivering comparable results across key metrics.

### 5.4.8 Testing Results on Imbalanced Data

The initial models were evaluated on the annotated testing data, focusing on their performance in handling class imbalances. The results for the traditional classifiers are shown in Fig. 5.5 and the results for the DL models are shown in Fig. 5.6.

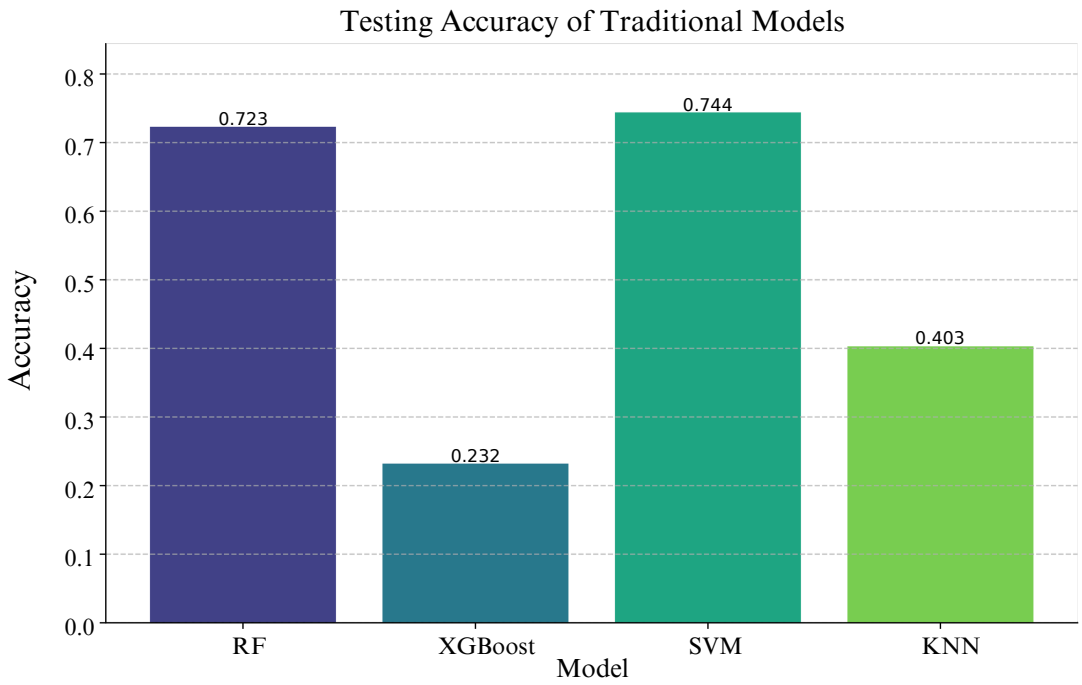



Figure 5.5: Testing results for traditional models

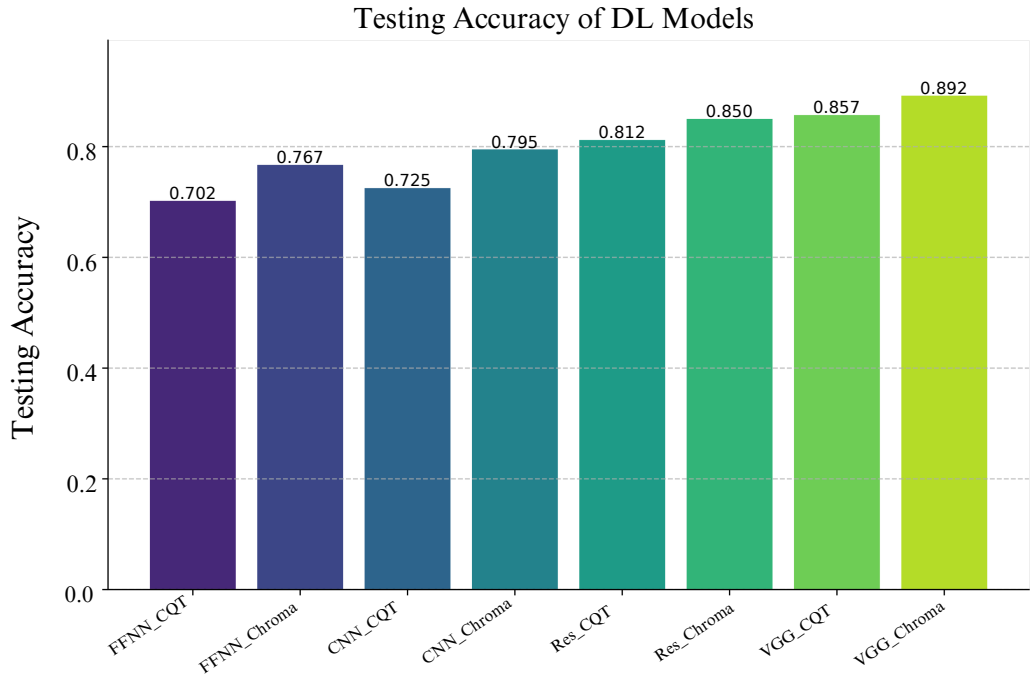


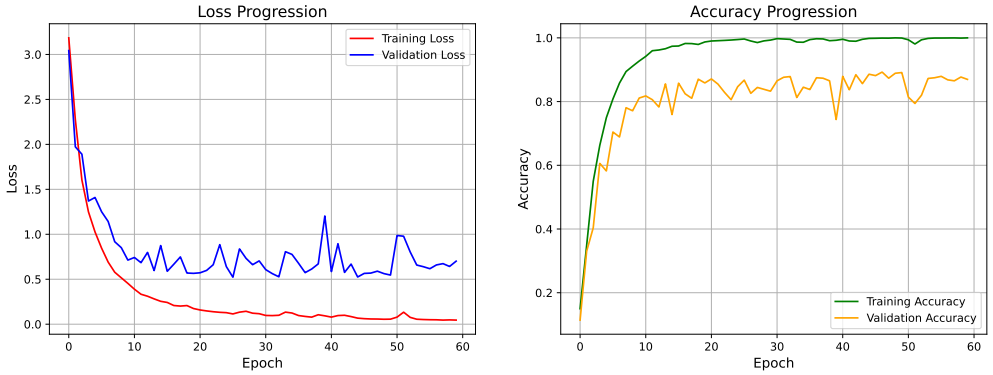

Figure 5.6: Testing results for DL models

## 5.5 Advanced Model Evaluation with Balanced Data

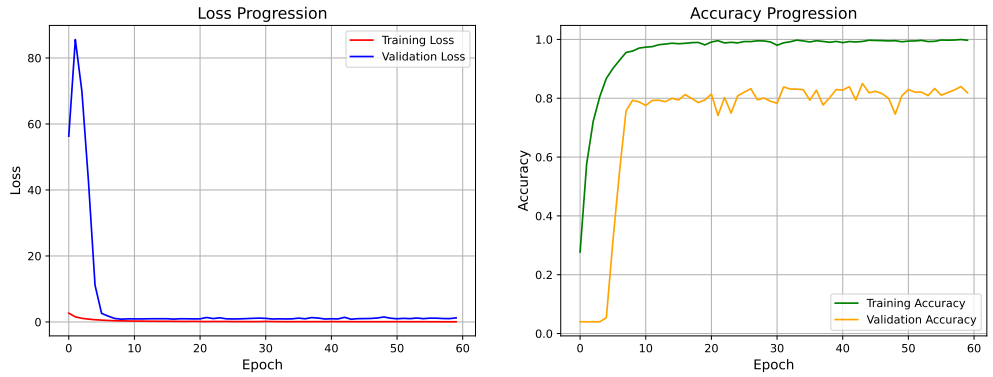
This section addresses the impact of balancing techniques on model performance and dataset characteristics. The dataset was balanced by oversampling minority classes to reach at least 70% of the samples present in the majority class. Table C.2 presents the results of this oversampling, demonstrating that the lowest sample count increased to 30 samples. The ratio between the most and least represented species decreased to 1.43:1.

### 5.5.1 Top Model Training Results

Due to their strong performance during initial imbalanced testing, this analysis focuses on two specific architectures: the VGG-16 model and the ResNet-50 model. Fig. 5.7 illustrates the training progression of the VGG-16 and ResNet architectures with their respective feature sets using a 3 second window.




(a) VGG-16 training progression on oversampled data




(b) ResNet-50 training progression on oversampled data

Figure 5.7: Training progression with the Mel-spectrogram-MFCC-chroma combination

To ensure a direct comparison with previous experiments, both models were trained using identical hyperparameters and optimisation settings. These two models are trained to also understand the impact of combining either CQT or chroma features with Mel-spectrograms and MFCCs.

The ResNet-50 initially exhibited a high model loss, peaking around 90 before stabilizing after approximately five epochs. In contrast, the VGG-16 began with a lower loss that stabilized around 1. This indicates that ResNet-50 faced more significant initial challenges in learning from the training data.

The validation accuracy for VGG-16 was 73.9% with a training accuracy of 97.2% using the chroma feature. When using CQT, the validation accuracy improved to 82.3%, with a training accuracy nearing 99.9%. For ResNet-50, the validation accuracy was slightly higher at 82.8%, with a perfect training accuracy of 100% using chroma. However, it dropped to 80.2% validation accuracy with CQT, also maintaining a 100% training accuracy.

The CQT-VGG-16 achieved an overall precision of 80% (macro average) and a weighted average of 81%. The recall values varied across classes, with notable weaknesses in classes 1 (0.55) and 4 (0.11), indicating difficulty in correctly identifying these classes. The F1-score showed a weighted average of 0.80, demonstrating balanced performance, but with room for improvement in underperforming classes.

The chroma-ResNet-50's overall precision was slightly higher, with a macro average of 80% and a weighted average of 84%. While it exhibited stronger recall values for certain classes, class 4 (0.17) was still notably low, similar to VGG-16. The weighted F1-score for ResNet-50 was 0.82, indicating better overall balance compared to VGG-16.

### 5.5.2 Top Model Testing Results

The models were evaluated using the same testing dataset as previously employed, with the addition of unannotated testing data. The testing results are shown in Fig. 5.8.

For the annotated testing data, ResNet with CQT achieved an accuracy of 76.7%. ResNet with chroma outperformed its CQT counterpart with an accuracy of 83.9%, highlighting the effectiveness of the chroma features in capturing relevant information. VGG-16 with chroma achieved the highest accuracy among all models at 86.5%, suggesting it is well-suited for this particular feature combination. The models' performance on

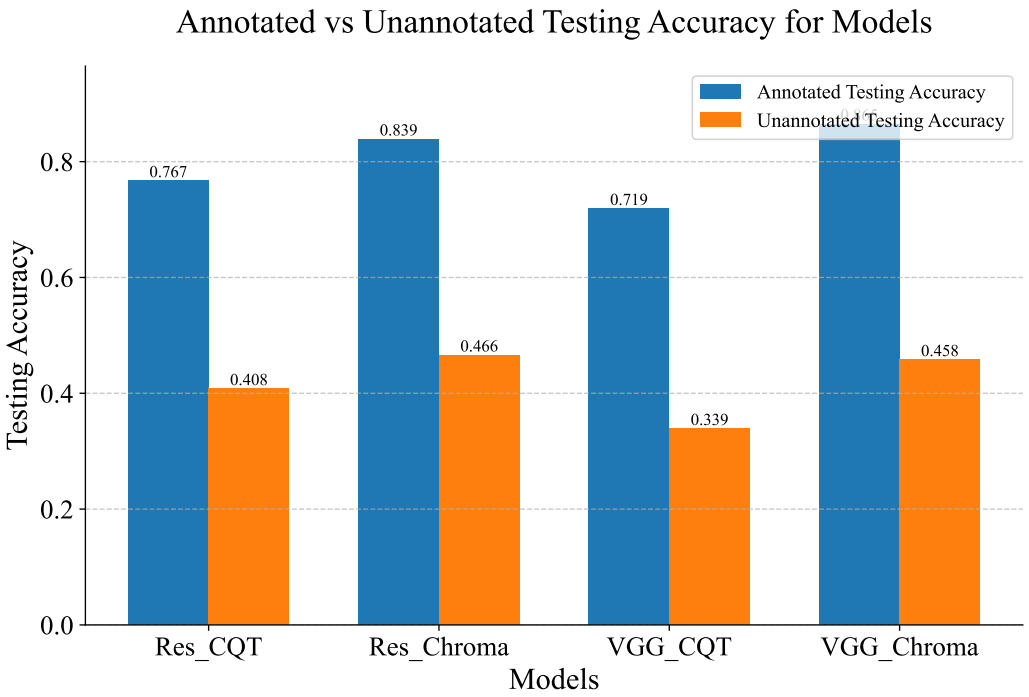



Figure 5.8: Performance comparison of VGG-16 and ResNet-50 models on the testing dataset, including unannotated data to simulate real-world conditions.

unannotated data revealed significant challenges in generalisation, considering the highest accuracy recorded with the unannotated data was 46.6%.

## 5.6 Comprehensive Performance Discussion

### 5.6.1 Imbalanced Data Evaluation

#### Traditional Model Performance

Among the traditional machine learning models, the RF emerged as the most effective on the test data. Although the XGBoost algorithm performed better during training, it failed to generalise effectively to unseen data. This highlights the importance of balancing both training and testing performance when evaluating models. Despite not achieving the 95.35% accuracy reported in the literature, the RF still stands as a strong candidate. Its testing performance, although slightly below expectations, is competitive given its quick training time (20 seconds) and fast prediction time (approximately 1 second). This makes it a highly efficient model for real-time applications where quick turnaround is crucial.

While the SVM showed higher testing accuracy than the RF, it comes with notable drawbacks. The training time for the SVM was around 3 minutes, and it required nearly 30 minutes to produce predictions for the testing data. This raises concerns about its scalability and practicality for larger datasets or real-time environments, despite its accuracy.

The KNN model, as a simple and intuitive algorithm, performed comparably to the results found in the literature. Although it is often used as a baseline in classification tasks, KNN demonstrated its ability to keep pace with more complex models, validating its role as a fundamental traditional machine learning technique.

The traditional models trained significantly faster when using averaged pooled features compared to the original 2D data. Moreover, these simplified features consistently outperformed their more complex counterparts in terms of accuracy, showcasing the models' preference for reduced dimensionality.

A common challenge for traditional models was their difficulty in classifying minority classes, particularly species that appeared less frequently in the training data. This was evident in the consistently low F1-scores for underrepresented species such as *Anas strepera*. These models tended to overfit to majority classes, struggling to generalise across the full range of bird species.

### Deep Learning Model Performance

In contrast to traditional models, deep learning approaches showed notable improvement in classifying minority species. Specifically, deep learning models demonstrated a better ability to distinguish underrepresented classes like *Anas strepera*, improving their F1-scores. This was due to the intrinsic architecture of neural networks, which allowed for the redistribution of weight emphasis across various classes.

The custom-designed CNN performed reasonably well, achieving a testing accuracy of 79.5%. However, when compared to transfer learning methods, there was a substantial gap. For instance, the VGG-16 model significantly outperformed the CNN, achieving a testing accuracy of 89.2%. This demonstrates the powerful advantage of leveraging pretrained models on large datasets to boost performance in more specific tasks like bird call classification.

The FFNN showed testing accuracies comparable to traditional models like SVM and

RF. However, due to the large number of epochs required, it took significantly longer to train, which may be a limiting factor in its practical applications.

The best-performing deep learning model, VGG-16, had a total of 14.8 million parameters (56.47 MB). Despite this large size, the model was capable of testing the entire dataset in just 4 seconds, showcasing both efficiency and effectiveness. The model's performance solidifies it as a leading choice for bird call classification in this study.

### 5.6.2 Feature Analysis

In traditional models, the averaged pooled features consistently outperformed the original 2D features. This simplification proved advantageous, suggesting that less complex representations of the audio data provided clearer signals for classification. The fact that the RF model achieved an accuracy above 70% with averaged features speaks to the effectiveness of this approach.

Single feature sets, such as CQT or chroma, consistently performed worse than combined feature sets. For imbalanced data, the best-performing feature combination was Mel spectrogram, MFCC, and CQT, while for balanced data, Mel spectrogram, MFCC, and Chroma delivered the highest accuracy.

MFCCs showed robustness across various bird calls and environmental conditions, consistently performing well even when used in isolation. This indicates that MFCCs are a reliable feature for bioacoustic classification, capable of capturing essential characteristics of bird vocalisations.

### 5.6.3 Balanced Data Evaluation

While balancing the dataset improved representation across species and led to higher F1-scores for minority classes, it did not necessarily improve the models' ability to generalise to new audio conditions. The augmentation techniques, designed to simulate real-world variations, did not significantly enhance the models' capacity to differentiate between bird calls more effectively than the imbalanced data.

The F1-scores for species that were previously underrepresented improved in the balanced dataset, demonstrating that balancing techniques are beneficial for fairer representation in model training. However, augmentation failed to capture the natural diversity

in bird vocalisations, particularly when considering the unique syllables and patterns of communication across different species.

The VGG-16 model performed best on annotated test data, while ResNet was better in unannotated test scenarios. However, overall results for field recordings remained low. The primary challenge was the model's inability to distinguish between background noise and actual bird calls, leading to misclassifications when background sounds were randomly assigned to bird species.

## 5.7 Chapter Summary

This chapter provides an extensive analysis of the performance of various machine learning and deep learning models applied to bird species classification. The evaluation process involved both imbalanced and balanced datasets, assessing how well models could generalise to unseen data, classify underrepresented species, and perform in challenging real-world conditions.

# Chapter 6

# Conclusions

## 6.1 Conclusions

The primary challenge addressed in this study is the accurate classification of bird species based on their audio recordings using ML techniques. Previous research has shown that while ML models have demonstrated strong performance on curated datasets, their generalisation to real-world audio conditions—characterised by background noise, recording variability, and overlapping bird calls—remains limited. The goal of this project was to bridge these gaps by evaluating traditional and deep learning models and exploring techniques to enhance their robustness in bird species classification.

This research successfully met the objectives outlined at the beginning of the study. Several key contributions include:

- A detailed survey of bird species classification using ML was conducted, establishing a solid foundation for the project.
- A variety of audio feature extraction methods were evaluated for their impact on classification accuracy.
- A thorough comparison between traditional ML models (RF, SVM, KNN) and deep learning models (CNN, FFNN, transfer learning with VGG-16 and ResNet-50) was carried out. The best performance was achieved using transfer learning with VGG-16, yielding an accuracy of 89.2% on testing data.
- The study demonstrated the benefit of data balancing techniques in improving

species representation, though it did not significantly enhance generalisation to new audio conditions.

- Augmentation methods like pitch shifting, noise injection, and time stretching were applied, but their impact on model robustness was limited by the complexities of real-world variations in bird vocalisations.

Due to time constraints and the availability of audio data, only 20 bird species were included in the dataset. This limits the generalisability of the findings to a broader range of species. The complexity of deep learning models, particularly when tuning hyperparameters, was constrained by the available computational power, even with the use of Google Colab Pro. More extensive training with larger datasets would require significantly more resources. A major limitation was the models' struggle to differentiate between bird calls and background noise, particularly in unannotated or field recordings. This highlights the need for more sophisticated noise handling techniques in future implementations.

The use of transfer learning with VGG-16 provided the most significant performance improvement. Its pre-trained architecture allowed the model to generalise well to unseen data, making it a highly effective approach for bird species classification. he combination of features such as Mel-spectrogram, MFCC, and Chroma consistently improved classification accuracy over single features. This supports the hypothesis that integrating multiple feature types captures a broader representation of the audio signal. Although outperformed by deep learning models, the RF model offered a strong baseline with an accuracy above 70% and near-instantaneous prediction time. Its simplicity and computational efficiency make it a valuable candidate for tasks where time is critical.

The insights gained from this research contribute significantly to the field of bioacoustic classification, particularly in the context of bird conservation. The application of transfer learning models has demonstrated that ML systems can approach real-time monitoring of bird populations, a critical need in biodiversity research. While most Acceptance Test Procedures (ATP) were successfully met, one key challenge remained: ensuring a minimum accuracy of 60% in models tested under realistic, noisy conditions, where the highest testing accuracy achieved was 46.6%. This limitation highlights the difficulty of generalising models to noisy environments, underscoring the need for further refinement.

Nonetheless, by improving model robustness and adaptability, this study supports the development of scalable, automated systems that can aid in monitoring endangered species, informing conservation strategies, and advancing ecological research.

## 6.2 Future Recommendations

As this project demonstrates the potential of ML techniques for bird species classification, several areas remain for further exploration to enhance performance and robustness. The following recommendations focus on addressing key challenges encountered during the study, such as improving model accuracy in noisy environments, expanding dataset diversity, and exploring advanced architectures like hybrid models and transformers.

### 6.2.1 Introducing a 'No Bird' Class

Based on the performance of current models and the challenges posed by background noise, an additional 'No Bird' class would significantly improve the ability of the system to distinguish between bird calls and non-bird sounds. In the annotated dataset, this could be implemented by identifying sections of the audio where bird calls are absent and labelling those as background noise. This would help train the model to better identify and ignore irrelevant audio segments, improving accuracy and reducing false positives. The literature has already highlighted noise as a persistent issue in bioacoustic classification, and the inclusion of this class could further address the challenge.

### 6.2.2 Handling Unannotated Data with Signal-to-Noise Techniques

For unannotated data, a multi-feature approach could be explored to detect bird calls based on signal-to-noise ratios. By synchronising features such as RMS Energy plots, CQT plots, and Mel Spectrograms, the system could identify sections of the audio with high signal-to-noise ratios, which likely correspond to bird calls. Expanding the window around these high-energy regions could help in more accurately labelling the bird call, while low signal-to-noise segments could be classified as noise. This method aligns with current literature recommendations that encourage feature-rich analysis.

### 6.2.3 Training on Larger and Diverse Datasets

To further test the robustness and generalisability of the models, expanding the training process to include new and larger datasets would be highly beneficial. Including more bird species, particularly those from diverse geographic regions, would help in assessing the model's ability to generalise to unfamiliar species. Additionally, incorporating field recordings from different environments (urban vs. rural, forest vs. wetland) could provide valuable insights into how well the models perform under varying conditions. This supports existing studies that recommend scalability for broader ecological applications.

### 6.2.4 Handling Overlapping Bird Calls

The current models show high AUC scores, suggesting they are effective at distinguishing between individual bird calls within a recording. However, to handle overlapping calls—an issue commonly encountered in real-world environments—parallel CNNs could be tested. These architectures could enable the system to process multiple audio streams simultaneously, effectively separating overlapping bird calls. Testing this approach would provide a more accurate reflection of the system's ability to perform in natural soundscapes, where multiple birds vocalise at the same time.

### 6.2.5 Improving Audio Augmentation

The dynamic bird call synthesiser mentioned in the literature review has shown promise in creating synthetic bird calls that simulate real-world conditions [41]. Implementing this technique could significantly improve the augmentation strategies used in this project. By generating synthetic variations of bird calls, the model would be exposed to a broader range of scenarios, improving its robustness to variations in bird vocalisations and reducing overfitting to specific audio patterns.

### 6.2.6 Exploring Unsupervised Learning for Limited Data

As highlighted in the literature, unsupervised learning models have the potential to greatly improve the classification process, particularly for species with limited available recordings. These models could automatically cluster similar-sounding bird calls, even

without labelled data, which would be invaluable for species that lack extensive datasets. Researchers and conservationists could then provide expert validation for these clustered groups, significantly enhancing the classification accuracy for rare or endangered species.

### 6.2.7 Real-Time Bird Call Monitoring System

After identifying the best-performing models, implementing a real-time bird call monitoring system could be the next step. This would allow for continuous monitoring of bird populations in natural environments, providing valuable data for conservation efforts and biodiversity studies. Testing this system under real-world conditions would provide a practical assessment of its performance, particularly in noisy and complex soundscapes.

### 6.2.8 Final Thoughts

By incorporating these future developments, the bird species classification system can become more robust, adaptable, and effective in real-world scenarios. Continued experimentation and refinement based on these recommendations will push the boundaries of automated bird call identification, contributing to the broader goals of environmental preservation and biodiversity monitoring.

# Bibliography


[1] A. Miyaguchi, N. Zhong, M. Gustineli, and C. Hayduk, "Transfer learning with semi-supervised dataset annotation for birdcall classification," 2023. [Online]. Available: https://arxiv.org/abs/2306.16760

[2] Pixabay Contributors, "Bird image," https://cdn.pixabay.com/photo/2016/03/02/13/59/bird-1232416_1280.png, 2016, accessed: 15 October 2024.

[3] Free Icons PNG Contributors, "Bird male humming bird png transparent background," https://www.freeiconspng.com/img/3503, 2024, accessed: 15 October 2024.

[4] Dorothymre, "Free icons png - bird transparent background - free transparent png download - pngkey," https://za.pinterest.com/pin/free-icons-png-bird-transparent-background-free-transparent-png-download-pngkey--7145946658665 2024, accessed: 15 October 2024.

[5] B. Chandu, A. Munikoti, K. S. Murthy, G. Murthy V., and C. Nagaraj, "Automated bird species identification using audio signal processing and neural networks," in *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, 2020, pp. 1–5.

[6] X. Han and J. Peng, "Bird sound classification based on ecoc-svm," *Applied Acoustics*, vol. 204, p. 109245, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003682X23000439

[7] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000273

[8] R. A. Bistel, A. Martinez, and G. B. Mindlin, "Neural networks that locate and identify birds through their songs," *European Physical Journal Special Topics*, vol. 231, pp. 185–194, 2022, received 20 July 2021; Accepted 16 December 2021; Published online 28 December 2021; © The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2021.

[9] J. Gómez-Gómez, E. Vidaña-Vila, and X. Sevillano, "Western mediterranean wetland birds dataset: A new annotated dataset for acoustic bird species classification," *Ecological Informatics*, vol. 75, p. 102014, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954123000432

[10] S. E. W. contributors, "Black-winged stilt — simple english wikipedia, the free encyclopedia," https://simple.wikipedia.org/wiki/Black-winged_stilt, 2024, accessed: 15 October 2024.

[11] L. Harper, "Bittern (botaurus stellaris) illustration," https://lizzieharper.co.uk/product/bittern-botaurus-stellaris/, 2024, accessed: 15 October 2024.

[12] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2018, open access.

[13] R. Gao, "Bird song classifier with machine learning," https://rachlllg.github.io/project/2023-Bird_Song_Classifier_with_Machine_Learning/#data-source, 2023, (Accessed on 10/04/2024).

[14] Y. Tang, C. Liu, and X. Yuan, "Recognition of bird species with birdsong records using machine learning methods," *PLOS ONE*, vol. 19, no. 2, p. e0297988, 2024.

[15] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A survey of audio classification using deep learning," *IEEE Access*, vol. 11, pp. 106 620–106 649, 2023.

[16] Y. Chen, Q. Guo, X. Liang, J. Wang, and Y. Qian, "Environmental sound classification with dilated convolutions," *Applied Acoustics*, vol. 148, pp. 123–132, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003682X18306121

[17] M. T. García-Ordás, S. Rubio-Martín, J. A. Benítez-Andrades, H. Alaiz-Moretón, and I. García-Rodríguez, "Multispecies bird sound recognition using a fully convolutional neural network," *Applied Intelligence*, vol. 53, pp. 23 287–23 300, July 2023, access provided by University of Cape Town.

[18] M. L. Clark, L. Salas, S. Baligar, C. A. Quinn, R. L. Snyder, D. Leland, W. Schackwitz, S. J. Goetz, and S. Newsam, "The effect of soundscape composition on bird vocalization classification in a citizen science biodiversity monitoring project," *Ecological Informatics*, vol. 75, p. 102065, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954123000948

[19] A. Naufal, P. Ehkan, L. Kamarudin, R. Kamarudin, and R. Ahmad, "A real-time portable bioacoustics species identification design concepts," *International Journal of Engineering and Technology (IJET)*, vol. 7, 05 2015.

[20] M. S. Imran, A. F. Rahman, S. Tanvir, H. H. Kadir, J. Iqbal, and M. Mostakim, "An analysis of audio classification techniques using deep learning architectures," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 805–812.

[21] S. Dian Handy Permana and K. Bayu Yogha Bintoro, "Implementation of constant-q transform (cqt) and mel spectrogram to converting bird's sound," pp. 52–56, 2021.

[22] T. Symbl, "Machine learning: Crash course in audio classification," https://symbl.ai/developers/blog/machine-learning-crash-course-in-audio-classification/, September 2022, (Accessed on 10/04/2024).

[23] E. Knight, K. Hannah, G. Foley, C. Scott, R. Brigham, and E. Bayne, "Recommendations for acoustic recognizer performance assessment with application to five common automated signal recognition programs," *Avian Conservation and Ecology*, vol. 12, 12 2017.

[24] S. Saha, "A comprehensive guide to convolutional neural networks — the eli5 way," https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, 2018, published in Towards Data Science, Accessed: 15 October 2024.

[25] S. Carvalho and E. F. Gomes, "Automatic classification of bird sounds: Using mfcc and mel spectrogram features with deep learning," *Vietnam Journal of Computer Science*, vol. 10, no. 01, pp. 39–54, 2023, accessed on 10/04/2024. [Online]. Available: https://doi.org/10.1142/S2196888822500300

[26] Q. Tang, L. Xu, B. Zheng, and C. He, "Transound: Hyper-head attention transformer for birds sound recognition," *Ecological Informatics*, vol. 75, p. 102001, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954123000304

[27] G. Gupta, M. Kshirsagar, M. Zhong, S. Gholami, and J. L. Ferres, "Comparing recurrent convolutional neural networks for large scale bird species classification," *Scientific Reports*, vol. 11, p. 17085, 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-96446-w

[28] S. Kahl *et al.*, "Birdnet: Ai-powered bird sound recognition," https://birdnet.cornell.edu/, 2024, accessed: 15 October 2024.

[29] T. M. Aide, C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega, and R. Alvarez, "Real-time bioacoustics monitoring and automated species identification," *PeerJ*, vol. 1, p. e103, 2013, highlighted in Top Animal Behavior Papers - November 2014; PeerJ Picks 2014 Collection.

[30] G. Boesch, "Very deep convolutional networks (vgg) essential guide," https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/, 2021, accessed: 2024-10-20. [Online]. Available: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Cornell University*, 2015, version 6, last revised 10 Apr 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015, tech report, Version 1. [Online]. Available: https://arxiv.org/abs/1512.03385

[33] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019, international Conference

on Machine Learning (ICML), Version 5. [Online]. Available: https://doi.org/10.48550/arXiv.1905.11946

[34] X.-C. Foundation, "Xeno-canto: Sharing bird sounds from around the world," https://xeno-canto.org/, accessed: 15 October 2024.

[35] I. C. P, D. K. R, and M. R, "Bird sound identification system using deep learning," *Procedia Computer Science*, vol. 233, pp. 597–603, 2024, 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050924006082

[36] "Macaulay library," https://www.macaulaylibrary.org/, accessed: 15 October 2024.

[37] J. Wang and G. Goldsztein, "Audio classification of bird species using convolutional neural networks," *Journal of Student Research High School Edition*, vol. 12, no. 1, 2023.

[38] F. Michaud, J. Sueur, M. Le Cesne, and S. Haupert, "Unsupervised classification to improve the quality of a bird song recording dataset," *Ecological Informatics*, vol. 74, p. 101952, 2023, institut Systématique, Évolution, Biodiversité (ISYEB), Muséum national d'Histoire naturelle, CNRS, Sorbonne Université, EPHE, Université des Antilles, 57 rue Cuvier, 75005 Paris, France. [Online]. Available: https://www.elsevier.com/locate/ecolinf

[39] Y. Maegawa, C. Haga, Y. Ushigome, T. Matsui, M. Suzuki, K. Taguchi, and K. Kobayashi, "A new survey method using convolutional neural networks for automatic classification of bird calls," *Ecological Informatics*, vol. 61, p. 101164, 2021. [Online]. Available: https://doi.org/10.1016/j.ecoinf.2021.101164

[40] C. Pérez-Granados, "Birdnet: applications, performance, pitfalls and future opportunities," *Ibis*, vol. 165, no. 3, pp. 1068–1075, 2023, open access.

[41] P. Tubaro and G. Mindlin, "A dynamical system as the source of augmentation in a deep learning problem," *Chaos, Solitons Fractals: X*, vol. 2, p. 100012, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590054419300107

[42] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022022, 2019, open access.

[43] Z. J. Ruff, D. B. Lesmeister, and C. L. Appel, "Workflow and convolutional neural network for automated identification of animal sounds," *Ecological Indicators*, vol. 124, p. 107419, 2021.

[44] G. Bota, R. Manzano-Rubio, L. Catalán, J. Gómez-Catasús, and C. Pérez-Granados, "Hearing to the unseen: Audiomoth and birdnet as a cheap and easy method for monitoring cryptic bird species," *Sensors*, vol. 23, no. 16, p. 7176, 2023, open access.

[45] E. Nemeth, N. Pieretti, S. A. Zollinger, N. Geberzahn, J. Partecke, A. C. Miranda, and H. Brumm, "Bird song and anthropogenic noise: vocal constraints may explain why birds sing higher-frequency songs in cities," *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1750, p. 20122798, 2013, received: 24 November 2012; Accepted: 13 December 2012; Subject Areas: behaviour, ecology; Keywords: anthropogenic noise, bird song, phonetogram, urbanization, blackbird, acoustic communication.

[46] K. McGinn, S. Kahl, M. Z. Peery, H. Klinck, and C. M. Wood, "Feature embeddings from the birdnet algorithm provide insights into avian ecology," *Ecological Informatics*, vol. 74, p. 101995, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954123000249

[47] C. Ozgur, T. Colliau, G. Rogers *et al.*, "MatLab vs. Python vs. R," *Journal of Data Science*, vol. 15, no. 3, pp. 355–372, 2017, published online: 4 August 2022. [Online]. Available: https://doi.org/10.6339/JDS.201707_15(3).0001

[48] A. Dyrcz, "Moustached warbler (*Acrocephalus melanopogon*), version 1.0," *Birds of the World*, 2020, text last updated November 23, 2013. [Online]. Available: https://birdsoftheworld.org/bow/species/mouwar1/cur/introduction?media=illustrations

[49] S. Vallath, "An ensemble approach to deep learning models in conversational intelligence," https://symbl.ai/developers/blog/an-ensemble-approach-to-deep-learning-models-in-conversational-intelligence/, March 2020, (Accessed on 10/04/2024).

[50] K. J. Gaston, "Birds and ecosystem services," *Current Biology*, vol. 32, no. 20, pp. R1163–R1166, 2022, accessed on 10/04/2024.

[51] P. H. Wightman, D. W. Henrichs, B. A. Collier, and M. J. Chamberlain, "Comparison of methods for automated identification of wild turkey gobbles," *Wildlife Society Bulletin*, vol. 46, no. 1, p. e1246, 2022, accessed on 10/04/2024.

[52] F. Adekogbe, "Introduction to classification algorithms," https://symbl.ai/developers/blog/custom-classifiers-audio-video-conversations-architecture/, May 2022, (Accessed on 10/04/2024).

[53] A. Thakur and P. Rajan, "Deep archetypal analysis based intermediate matching kernel for bioacoustic classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 298–309, 2019.

[54] J. Burk, L. Polansky, P. Repetto, L. Roberts, and D. Rockmore, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/02_02.php, August 2022, (Accessed on 10/04/2024).

[55] ——, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/02_04.php, August 2022, (Accessed on 10/04/2024).

[56] ——, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/02_08.php, August 2022, (Accessed on 10/04/2024).

[57] ——, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/03_01.php, August 2022, (Accessed on 10/04/2024).

[58] ——, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/02_03.php, August 2022, (Accessed on 10/04/2024).

[59] ——, "Chapter 2: The digital representation of sound, part two: Playing by the numbers," https://musicandcomputersbook.com/chapter2/02_01.php, August 2022, (Accessed on 10/04/2024).

[60] Y. Maegawa, Y. Ushigome, M. Suzuki, K. Taguchi, K. Kobayashi, C. Haga, and T. Matsui, "A new survey method using convolutional neural networks for automatic classification of bird calls," *Ecological Informatics*, vol. 61, p. 101164, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157495412030114X

[61] A. C. Lees, L. Haskell, T. Allinson, S. B. Bezeng, I. J. Burfield, L. M. Renjifo, K. V. Rosenberg, A. Viswanathan, and S. H. M. Butchart, "State of the world's birds," *Annual Review of Environment and Resources*, vol. 47, pp. 231–260, 2022, first published as a Review in Advance on May 4, 2022.

[62] M. Arif, R. Hedley, and E. Bayne, "Testing the accuracy of a birdnet, automatic bird song classifier," 2020, conference/Workshop Poster. [Online]. Available: https://era.library.ualberta.ca/items/42690fd9-7e50-4534-8361-cefbf7f40b49

[63] K. Riebel, "Understanding sex differences in form and function of bird song: The importance of studying song learning processes," *Frontiers in Ecology and Evolution*, vol. 4, June 2016, this article is part of the Research Topic: Fitness Costs and Benefits of Female Song. [Online]. Available: https://doi.org/10.3389/fevo.2016.00062

[64] C. J. Whelan, H. Şekercioglu, and D. G. Wenny, "Why birds matter: from economic ornithology to ecosystem services," *Journal of Ornithology*, vol. 156, no. Suppl 1, pp. S227–S238, 2015.

[65] "xkcd: Thesis defense," https://xkcd.com/1403/, (Accessed on 10/20/2017).

[66] "Big bang - warm kitty, soft kitty (sheldon's lullaby sick song) instrumental version lyrics — metrolyrics," http://www.metrolyrics.com/warm-kitty-soft-kitty-sheldons-lullaby-sick-song-instrumental-version-lyrics-big-bang.html?ModPagespeed=noscript, (Accessed on 10/20/2017).

[67] M. Shaw, "Writing good software engineering research papers," in *Software Engineering, 2003. Proceedings. 25th International Conference on.* IEEE, 2003, pp. 726–736.

[68] B. Paltridge, "Thesis and dissertation writing: an examination of published advice and actual practice," *English for Specific Purposes*, vol. 21, no. 2, pp. 125–143, 2002.

[69] U. Eco, *How to write a thesis*. MIT Press, 2015.

[70] I. J. Lovette and J. W. Fitzpatrick, *Handbook of Bird Biology*. John Wiley Sons, Jun 2016. [Online]. Available: https://www.wiley.com/en-us/Handbook+of+Bird+Biology%2C+3rd+Edition-p-9781118291054

[71] L. Ilaria, *Birds and Flowers: Colors of the Sky and the Earth*. Independently published, Jun 2024.

[72] 2024. [Online]. Available: https://www.iucnredlist.org

[73] S. A. Z. N. G. J. P. A. C. M. Erwin Nemeth, Nadia Pieretti and H. Brumm, "Bird song and anthropogenic noise: vocal constraints may explain why birds sing higher-frequency songs in cities," *Proceedings of the Royal Society B: Biological Sciences*, 2013.

[74] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A survey of audio classification using deep learning," *IEEE Access*, vol. 11, pp. 106 620–106 649, 2023.

# Appendix A

# Graduate Attribute Tracking

| Graduate Attribute | Response |
|---|---|
| **GA 1: Problem Solving** | I demonstrated problem-solving skills by tackling the challenge of accurately classifying bird species from audio recordings, addressing issues like imbalanced data and background noise. I researched and selected machine learning techniques through a thorough literature review, then experimented with various architectures to solve the classification problem. By applying principles from natural sciences, mathematics, and engineering to data preprocessing, I enhanced model performance. My problem-solving approach evolved as I refined models, compared results, and reached evidence-based conclusions on the most effective techniques. This solution has practical value for ornithologists and birdwatchers, automating audio analysis and allowing them to focus more on fieldwork and conservation efforts. |
| **GA 4: Investigations, Experiments, and Data Analysis** | I demonstrated skills in investigations, experiments, and data analysis by developing a systematic approach to bird audio classification. This involved engaging with the latest research on audio classification and machine learning, ensuring I applied state-of-the-art techniques. I designed and conducted experiments, focusing on critical steps like audio preprocessing, feature extraction, and comparing machine learning models, including deep learning versus traditional methods. Each experiment was carefully documented, outlining methodologies, parameters, and outcomes. I analysed key metrics like accuracy, precision, and recall to evaluate model performance and applied data augmentation techniques to test their impact on generalisation. By refining models through iterative experiments and performing comprehensive data analysis, I validated the accuracy of the algorithms and drew evidence-based conclusions on the most effective classification methods. This thorough approach informed deeper insights into the effectiveness of various techniques and their real-world applicability for conservation efforts. |

| | |
|---|---|
| **GA 5: Use of Engineering Tools** | I demonstrated effective use of engineering tools throughout the development, implementation, and analysis phases of my bird audio classification project. Using Python as the core language, I leveraged libraries like TensorFlow to implement machine learning models, especially Convolutional Neural Networks (CNNs), and Librosa for audio processing and feature extraction. I utilized Visual Studio Code (VSCode) as the integrated development environment for coding, debugging, and optimizing algorithms. For data analysis and visualisation, I employed tools like Matplotlib and Pandas to generate performance metrics, plots, and tables, facilitating model comparisons. Additionally, I used Git and GitHub for version control to manage code iterations and track progress. My thorough application of these tools allowed for seamless execution and analysis of experiments, from preprocessing to model evaluation. |
| **GA 6: Professional and Technical Communication** | I demonstrated strong professional and technical communication skills throughout my work on bird audio classification. I engaged with current research literature, designed experiments to compare machine learning models, and implemented data analysis techniques to interpret results. By synthesizing this information, I reached well-founded conclusions on model performance. I balanced theoretical research with practical experimentation, applying a rigorous research methodology to deepen understanding of audio classification. In weekly meetings with my supervisor, I clearly communicated progress and findings. I documented these results in a comprehensive report following IEEE referencing standards, including detailed explanations of machine learning architectures, preprocessing techniques, and results. Visual aids like graphs and tables were used to effectively present complex data. My literature review critically analysed relevant research, and I plan to further demonstrate my communication skills through a presentation and poster to showcase the outcomes of my work. |
| **GA 8: Individual Working** | I demonstrated strong individual working skills by independently managing every aspect of my bird audio classification project, from initial research and problem definition to implementation and analysis. With minimal guidance from my supervisor, I took the initiative to select appropriate machine learning techniques, design experiments, and critically analyse results. This required self-motivation, effective time management, and informed decision-making at each stage. I maintained steady progress by setting personal deadlines, prioritizing tasks, and working autonomously, while also knowing when to seek expert advice. My ability to balance independent work with professional ethics highlights my development of essential individual working skills. |

| | |
|---|---|
| **GA 9: Independent Learning Ability** | I demonstrated strong independent learning ability, essential for professional growth in engineering, through self-directed research and mastery of advanced concepts in machine learning and audio processing. I independently studied various Python libraries and navigated complex technical documentation, learning how to implement Mel spectrograms and neural network architectures. This project required adapting to new tools and programming paradigms, which I accomplished through focused self-study. By directly referencing technical documentation and applying it in practice, I showcased flexibility and an ability to learn and apply new frameworks effectively, reinforcing my independent learning capabilities. |

# Appendix B

# Use of AI Tools for Assistance

In this project, AI tools were employed for two main purposes: debugging code and improving grammar. Below are examples illustrating how AI was used to assist in these tasks.

## B.1 Code Assistance Example

**Prompt:** Help me debug this Python code for extracting Mel-Spectrograms.

**Input:**

```
import librosa
audio, sr = librosa.load('audio.wav')
mel_spec = librosa.feature.melspectrogram(y=audio, sr=sr)
plt.imshow(mel_spec)
```

**AI Output:**

The `librosa.feature.melspectrogram` function returns a 2D array, which needs to be converted to decibels before visualizing.

**Corrected Code:**

```
import librosa
import librosa.display
audio, sr = librosa.load('audio.wav')
mel_spec = librosa.feature.melspectrogram(y=audio, sr=sr)
```

```
mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)
librosa.display.specshow(mel_spec_db, sr=sr, x_axis='time', y_axis='mel')
```

## B.2 Grammar Assistance Example

**Prompt:** Check and improve the grammar in the following sentence. **Input:** "The results shows a clear correlation between the features extracted and the species predicted."

**AI Output:** "The results show a clear correlation between the extracted features and the predicted species."

These examples illustrate how AI tools were used to assist in debugging code and enhancing the grammatical quality of the report. All technical content, analysis, and core ideas remain my own work, while AI tools were employed for support in these auxiliary tasks.

# Appendix C

# Exploratory Data Analysis Of Dataset

This appendix provides supplementary visualisations and details from the Exploratory Data Analysis (EDA) phase of the implementation. It contains figures and tables derived from the Western Mediterranean Wetland Birds (WMWB) dataset.

## C.1 Imbalanced Species Composition

The imbalanced distribution of samples per species is shown in Tab. C.1 and the audio duration distribution per species is shown in Fig. C.1.

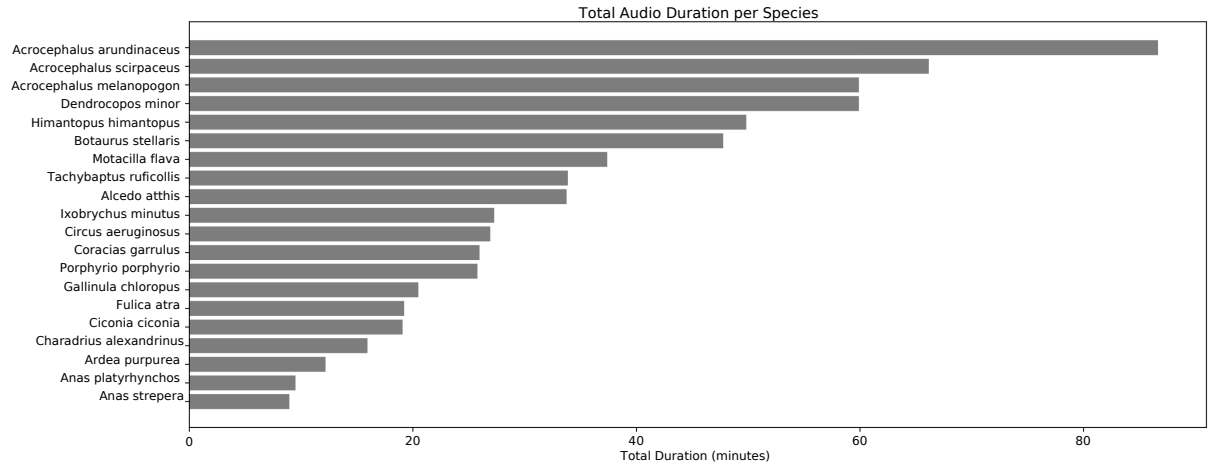



Figure C.1: Distribution of duration per species

| Encoded Class | Species | Number of Samples |
|---|---|---|
| 0 | Acrocephalus arundinaceus | 34 |
| 1 | Acrocephalus melanopogon | 50 |
| 2 | Acrocephalus scirpaceus | 37 |
| 3 | Alcedo atthis | 64 |
| 4 | Anas platyrhynchos | 19 |
| 5 | Anas strepera | 9 |
| 6 | Ardea purpurea | 43 |
| 7 | Botaurus stellaris | 56 |
| 8 | Charadrius alexandrinus | 58 |
| 9 | Ciconia ciconia | 40 |
| 10 | Circus aeruginosus | 38 |
| 11 | Coracias garrulus | 24 |
| 12 | Dendrocopos minor | 39 |
| 13 | Fulica atra | 55 |
| 14 | Gallinula chloropus | 54 |
| 15 | Himantopus himantopus | 70 |
| 16 | Ixobrychus minutus | 38 |
| 17 | Motacilla flava | 42 |
| 18 | Porphyrio porphyrio | 53 |
| 19 | Tachybaptus ruficollis | 56 |
| | **Total** | **879** |

Table C.1: Species and Sample Counts

## C.2 Training and Validation Set Composition

The dataset was split into a 80/20 train-validation split. The distribution of duration per species in the training and validation sets is as follows:

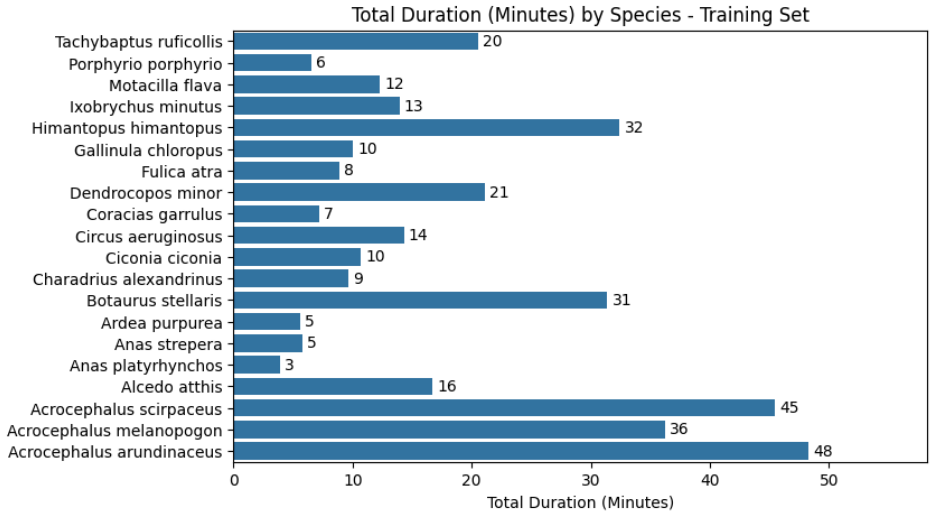



Figure C.2: Distribution of training duration per species

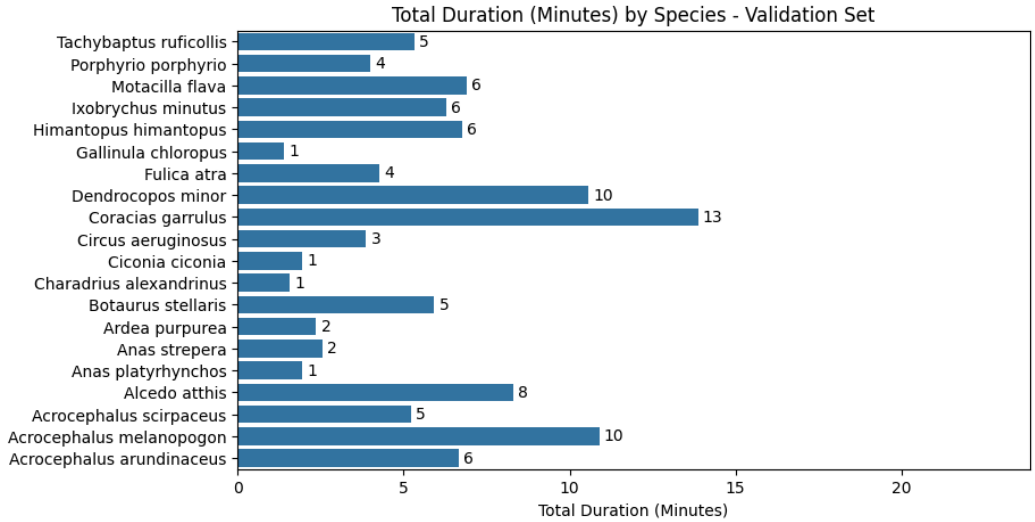



Figure C.3: Distribution of training duration per species

These visualisations provide a comprehensive overview of the dataset's species composition and the distribution of samples across the training and validation sets, highlighting the challenges posed by the imbalanced nature of the data.

## C.3 Balanced Species Composition

The imbalanced distribution of samples per species is shown in Tab. C.2.

| Encoded Class | Species | Number of Samples |
|---|---|---|
| 0 | Himantopus himantopus | 43 |
| 1 | Alcedo atthis | 37 |
| 2 | Tachybaptus ruficollis | 36 |
| 3 | Botaurus stellaris | 32 |
| 4 | Circus aeruginosus | 30 |
| 5 | Anas platyrhynchos | 30 |
| 6 | Coracias garrulus | 30 |
| 7 | Dendrocopos minor | 30 |
| 8 | Ciconia ciconia | 30 |
| 9 | Ixobrychus minutus | 30 |
| 10 | Acrocephalus arundinaceus | 30 |
| 11 | Motacilla flava | 30 |
| 12 | Acrocephalus scirpaceus | 30 |
| 13 | Ardea purpurea | 30 |
| 14 | Acrocephalus melanopogon | 30 |
| 15 | Fulica atra | 30 |
| 16 | Porphyrio porphyrio | 30 |
| 17 | Charadrius alexandrinus | 30 |
| 18 | Gallinula chloropus | 30 |
| 19 | Anas strepera | 30 |

Table C.2: Species and Sample Counts after Oversampling

# Appendix D

# Code and Results Repository

This section provides a reference to the complete codebase associated with this project. The repository contains all essential scripts, algorithms, and methods used for data processing, feature extraction, model development, and supporting functions. By sharing this code, the goal is to ensure transparency, reproducibility, and facilitate further exploration of the techniques employed in the project.

For access to the full code and results, please visit the GitHub repository:
https://github.com/Thato-Mot/Final-Version-Of-Bird-Classification-Project

## D.1 Traditional Model Performance Based On Imbalanced Data

This appendix summarises the performance data of traditional models trained for bird species classification based on the initial imbalanced data.

### D.1.1 Random Forest (RF) Model

Tab. D.1 shows the results of the hyperparameter tuning done on the Random Forest model.

Table D.1: Random Forest Hyperparameter Tuning Results

| No. Estimators | Criterion | Balanced | Train Accuracy | Val Accuracy |
|---|---|---|---|---|
| 50 | Entropy | No | 1.0 | 0.650 |
| 100 | Entropy | Yes | 1.0 | 0.640 |
| 200 | Entropy | Yes | 1.0 | 0.667 |
| 100 | Gini | No | 1.0 | 0.664 |
| 100 | Log Loss | No | 1.0 | 0.659 |
| 200 | Log Loss | No | 1.0 | 0.670 |
| 200 | Log Loss | Yes | 1.0 | 0.653 |

### D.1.2 XGBoost Model

Tab. D.2 shows the results of the hyperparameter tuning done on the Random Forest model.

Table D.2: XGBoost Hyperparameter Tuning Results

| Estimators | Booster | Learning Rate | Train Accuracy | Val Accuracy |
|---|---|---|---|---|
| 100 | `dart` | 0.25 | 1.0 | 0.661 |
| 100 | `gblinear` | None | 1.0 | 0.651 |
| 300 | `gblinear` | 0.50 | 1.0 | 0.647 |
| 100 | `gbtree` | 0.25 | 1.0 | 0.676 |
| 100 | `gbtree` | None | 1.0 | 0.688 |
| 100 | `gbtree` | None | 1.0 | 0.668 |

### D.1.3 Support Vector Machine (SVM) Model

**Hyperparameter Adjustments**

The key parameters adjusted include the regularisation parameter $C$, the kernel coefficient $\gamma$, and the choice of kernel.

The $C$ parameter controls the trade-off between achieving a low training error and a low testing error, with larger values emphasizing the reduction of training error. The values tested for $C$ were: 0.5, 4, 6, 10, 100, 150, 200, and 300. A comprehensive range

was selected to assess the impact of this parameter on the model's ability to generalize to unseen data.

For the $\gamma$ parameter, which defines the influence of individual training examples, a selection of values was examined: 'scale', 1, 0.1, 0.01, 0.001, 1.5, 2, 2.5, and 3. This range allows for an evaluation of how different scales of influence affect model performance, particularly in relation to the complexity of the decision boundary.

To ensure robustness in performance evaluation, a five-fold cross-validation (CV = 5) strategy will be employed. This technique, repeated five times, involves partitioning the training dataset into five subsets, where the model is trained on four subsets and validated on the remaining one.

## D.2 DL Model Performance Based On Imbalanced Data

### D.2.1 Feedforward Neural Network (FFNN) Model

Table D.3 summarises the hyperparameter tuning results.

Table D.3: FFNN Hyperparameter Tuning Results

| Hidden Layers | Epochs | Learning Rate | L2 Reg. | Dropout | Training Accuracy | Valida. Accuracy |
|---|---|---|---|---|---|---|
| [128, 64, 32] | 50 | 0.00010 | 0.0015 | 0.3 | 0.860 | 0.720 |
| [256, 128, 64] | 60 | 0.00010 | 0.0020 | 0.4 | 0.884 | 0.744 |
| [256, 128, 64] | 100 | 0.00010 | 0.0020 | 0.4 | 0.946 | 0.717 |
| [128, 64, 64] | 100 | 0.00001 | 0.0005 | 0.5 | 0.892 | 0.672 |
| [128, 128, 64] | 90 | 0.00003 | 0.0001 | 0.4 | 0.917 | 0.729 |
| [64, 64, 32] | 100 | 0.00010 | 0.0020 | 0.4 | 0.951 | 0.677 |
| [256, 128] | 70 | 0.00010 | 0.0012 | 0.4 | 0.933 | 0.723 |

### D.2.2 Convolutional Neural Network (CNN) Model

Table D.4 summarises the hyperparameter tuning results.

Table D.4: CNN Hyperparameter Tuning Results

| Conv Layers | Dense Units | Learning Rate | L2 Reg. | Conv Dropout | FC Dropout | Best Epoch | Val Acc |
|---|---|---|---|---|---|---|---|
| [64, 32] | [256] | 0.00005 | 0.15 | 0.30 | None | 100 | 0.681 |
| [64, 32, 32] | [256, 128] | 0.00040 | 0.25 | 0.40 | 0.10 | 70 | 0.707 |
| [32, 32] | [128] | 0.00010 | 0.20 | 0.30 | 0.20 | 90 | 0.738 |
| [64, 32, 32] | [256, 64] | 0.00002 | 0.05 | 0.25 | 0.30 | 100 | 0.716 |
| [64, 32, 16] | [256, 128] | 0.00030 | 0.3 | 0.50 | 0.20 | 100 | 0.685 |