DEPARTMENT OF ENGINEERING MATHEMATICS

Birdsong Classification with Wavelet Transforms

and Deep Learning Methods

Anthony Roan

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

Friday 1st September, 2023

Supervisor: Dr Martin Homer

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MSc in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Anthony Roan, Friday 1st September, 2023

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

The BirdCLEF Kaggle competition plays a pivotal role in avian conservation science by challenging participants to identify bird species based on their calls. While the competition has seen significant advancements, including various models and techniques, gaps remain in achieving high classification accuracy. This research addresses these gaps by introducing a novel method combining Continuous Wavelet Transform (CWT) with noise reduction and Convolutional Neural Networks (CNNs) to enhance bird-call classification.

The study employs a meticulously designed preprocessing signal pipeline to improve the audio quality of competition data and reduce the background noise present in wildlife recordings, all without manual curation or selection of raw audio data. The bird song data is transformed using CWT, and a thresholding technique is explored for postprocessing noise reduction and computational efficiency. This research not only underscores the advantages of CWT over the more commonly used short-time Fourier transforms (STFT) but also visually demonstrates the effectiveness of the preprocessing pipeline techniques.

Key contributions of this research include:

- A comprehensive review of BirdCLEF history, entries, and classification methods, focusing on the mathematical techniques underlying bird species classification.
- The design and implementation of an efficient signal processing pipeline featuring effective noise reduction and segmentation techniques.
- The development of a TensorFlow GPU pipeline and the training of a CNN model that achieved superior classification accuracy.
- The development of a traditional classifier, using features defined in published literature and competition winners, for comparative analysis with the CNN model.

The insights provided by this research and the proposed method's enhanced performance hold promise for influencing future BirdCLEF competitions and contributing to advancements in the related sciences.

All code available at: https://uob-my.sharepoint.com/:f:/g/personal/ww22896_bristol_ac_uk/EqzjgKDJ0ptAttpQPKP24HEBpvVU2cVFAF4x2EnoPIN3wg?e=1Wfrg0.

Ethics statement: This project does not require ethics approval, as reviewed by my supervisor, Dr Martin Homer, Associate Professor in Mathematical Modelling.

I have completed the ethics test on Blackboard. My score is 12/12.

Supporting Technologies

- Code was developed using a recent version of Python 3, using data science libraries such as NumPy and Pandas in a Virtual Studio environment.
- All the data processed in this study can be found in the BirdClef 2023 Training data [31].
- Google Collaboratory for online GPU resources and notebook-style code execution [28].
- Models were developed, tested and trained using Tensorflow & Keras.
- Essential packages included Pandas 1.4.4 for data manipulation, Matplotlib 3.7.1 and Seaborn 0.12.2 for visualisation, and Scikit-Learn 1.30 for machine learning.
- Signal processing relied on the functions contained in the pyAudioAnalysis package [25].
- LaTeX format for this thesis, via the online service *Overleaf*.
- random_state is always 42.

Notation and Acronyms

AI	:	Artificial Intelligence
BTO	:	British Trust for Ornithology
CNN	:	Convolutional Neural Network
CWT	:	Continuous Wavelet Transform
CWT-CNN	:	Continuous Wavelet Transform-based CNN
DAT	:	Digital Audio Tape
DWT	:	Discrete Wavelet Transform
FFT	:	Fast Fourier Transform
GPU	:	Graphics Processing Unit
JPEG	:	Joint Photographic Experts Group
mAP	:	Mean Average Precision
MFCCs	:	Mel-Frequency Cepstral Coefficients
Apps	:	Mobile Applications
PAM	:	Passive Acoustic Monitoring
SNR	:	Signal to Noise Ratio
STFT	:	Short Time Fourier Transform
TPU	:	Tensor Processing Unit
VAD	:	Voice Activity Detection
XC	:	Xeno-Canto
XGB	:	Gradient Boost Classifier

Acknowledgements

This work is wholeheartedly dedicated to my former teachers and classmates at St Alban's Catholic Primary School in Tremorfa, Cardiff. I especially want to thank Miss Donnelly, Mrs Cunningham, Mrs Collins, Mr Mansfield, Mike Rein, Tim Britton, and the late, great John Harrington, "eyes as big as saucers."

School was a fascinating place for me, made special by caring teachers and friends, whom I still cherish today. Without the solid educational foundation and encouragement I received there, I would not be returning to academia after nearly forty years.

To all my St Alban's teachers, friends, and members of the Parish, I sincerely thank you.

"Quanto plus scio, tanto magis intellego me nescire."

Chapter 1

Introduction

The BirdCLEF 2023 competition challenges data scientists to identify bird species from audio recordings collected in Kenya. Conservation scientists consider monitoring avian populations a severe matter, as birds are susceptible to environmental changes and can be early warning systems for the health of an ecosystem. Bird populations, spread across all terrestrial habitats, make them an effective indicator of biodiversity [50].

BirdClef aims to develop accurate classification solutions to enable the monitoring of populations to assist worldwide restoration projects. Conservationists require affordable and efficient tracking of population changes, especially in rugged terrain and vast areas. This is crucial for assessing global restoration projects, where traditional manual surveys are costly and challenging to manage [59].

The Natural State Research Centre uses bioacoustic monitoring and Artificial Intelligence (AI) to track their restoration impact on Northern Mount Kenya's biodiversity project. By developing scalable monitoring technology, Natural State aims to 'catalyze' investment in conservation projects focused on sustainable land management and wildlife protection [74].

1.1 Birdsong

Fascination with birdsong dates back to ancient civilisations. The ancient Greeks attributed divine qualities to these sounds, as discussed by Socrates in Plato's Phaedo [36] around 360 B.C.E.:

"But men, because they are themselves afraid of death, slanderously affirm of the swans that they sing a lament at last, not considering that no bird sings when cold, or hungry, or in pain, not even the nightingale, nor the swallow, nor yet the hoopoe; which are said indeed to tune a lay of sorrow."

In 1889, the naturalist Ludwig Koch used an Edison phonograph gifted by his father to record the song of captive white-rumped shama, marking the first known bird song recording [76]. By the 1930s, Koch had established himself as a leader in recording nature sounds, mainly through his collaboration with Sir Julian Huxley. Capturing wild birds in their natural habitats presented numerous challenges. Still, their joint efforts resulted in the 1938 publication of "Songs of Wild Birds", allowing people to experience authentic birdsong at home [35].

The advent of recording technology in the 20th century transformed the collection of birdsong. Early mechanical imitations gave way to actual field recordings of birds in their natural habitats, enabling more accurate representation and preservation of birdsong. Technological advances drove the new techniques: in 1934, an emperor penguin was the first bird recorded via radio transmission from the field to a receiver connected to a recorder, enabling remote recording [7].

Magnetic tape recorders were first used to capture birdsong in 1946 by Sture Palmér in Sweden [11], following which portable reel-to-reel recorders in the 1960s enabled longer, higher-quality recordings. An increase in ornithological interest has even led to the rediscovery of extinct bird species. The Puerto Rican Whip-poor-will was rediscovered in 1961 after locals recorded its calls, confirming its existence and allowing for documentation of its unique song [62].

Digital recording technologies that emerged in the 1980s, such as DAT (digital audio tape), revolutionised the documentation of birdsong by enabling extended frequency ranges and low self-noise devices. DAT offered advantages for field recordings and was later followed by analogue and digital cassette for-

mats, compact disc and MiniDisc. Current digital systems with hard disk and Flash memory provide brilliant audio quality, simple editing and analysis, easy sharing, and vast storage capacity [55].

These advances have resulted in a prolific rise in the recording of avian songs. The British Library's Sound Archive contains an extensive collection of birdsong recordings [38]. In 1965, only 25% of known bird species had been recorded, but this figure rose to over 90% today. The archive houses commercial, private, and historical recordings encompassing over 80% of the world's avian species. The birdsong collection offers value as a resource for research and enjoyment.

1.2 BirdClef

Birds are excellent indicators of biodiversity change due to their mobility and diverse habitats. While effective, observer-based bird surveys are logistically challenging and expensive. Passive acoustic monitoring (PAM), when combined with machine learning tools, offers a more scalable and cost-effective solution [91]. The BirdCLEF competition aims to harness the power of machine learning to process continuous audio data and recognise bird species by their calls. The ultimate goal is to aid conservation efforts by providing accurate tools for monitoring avian biodiversity.

BirdCLEF rewards the best contributions and, for the 2023 edition, offered a share of $50,000 for the first five places. Additionally, participants are encouraged to submit working notes to the LifeCLEF 2023 conference, where they may be given a special award, "Best BirdCLEF Working Note," with each winner receiving $2,500.

LifeCLEF, now called BirdCLEF, launched its first bird task in 2014, aiming to identify 501 bird species from Brazil. The dataset consisted of 14,027 recordings of 501 species taken from Xeno-Canto's collaborative database and included at least 15 recordings per species, reflecting real-world variations in equipment and noise. Ten groups participated, submitting 29 runs. To evaluate the success of identifying multiple species in a recording, the mean average precision (mAP) metric was used. This metric is handy for tasks where multiple object classes overlap, such as when two bird species call simultaneously. The highest-performing model achieved 0.511 mAP, while the lowest attained 0.002 mAP [39].

The competition has evolved, adapting to the latest technological advancements. Introducing deep learning brought drastic improvements in mAP scores; for example, the best mAP score improved in 2016 when deep learning, including CNNs, were first applied. Results showed CNNs outperformed other approaches, with the top system achieving 0.69 mAP on the 2015 test set [27]. By 2023, the top 5 solutions had similar scores of 0.76. These solutions blended audio processing and computer vision techniques, emphasising data quality and preprocessing. The methods included the use of ensembling and pre-trained models [30].

The BirdCLEF competition plays a pivotal role in advancing the field of automatic birdsong identification. Its rigorous challenges and significant rewards have made it a cornerstone event for researchers, data science practitioners and hobbyists alike.

1.2.1 Leading Solutions

BirdCLEF 2023 showcases a range of innovative approaches to bird song identification, with deep learning methods playing a pivotal role in the most successful entries, which employed one or more of the following methods:

- Pre-training: One leading strategy involved the pre-training of models on BirdCLEF datasets from previous years (2020, 2021, and 2022) and the Xeno-Canto Extend dataset [61], followed by fine-tuning on the 2023 data [4], significantly enhancing model performance.
- Spectrogram: A popular Fourier transform approach that involves changing the audio data into spectrograms and then training a deep learning model on the processed data [72].
- Data Augmentation: Using EfficientNet, a CNN-type model for pre-training, and data augmentation methods like CutMixUp are also particularly effective. These techniques improve the model's performance by adding variety to the training data [3].

Despite the success of existing methods, the BirdCLEF competition remains fiercely contested, with competitors continually seeking the slightest advantage. They do so by exploring innovative variations on familiar themes, and this competitive landscape underscores the innovation potential. With a mere 0.89% difference between the first and fifth place, exploring new avenues could be the key to future success.

1.2.2 Competition Data

The competition organisers provide training data, including brief recordings of individual bird calls contributed by Xeno-canto (XC) users. In 2023, there are approximately 16.9k sample files covering a range of 264 species. Accompanying the training data is a range of metadata, with fields including the primary label (a bird species code), latitude and longitude (the coordinates of the recording location), author (the user who contributed the recording), and filename. The competition employs a hidden test set, with submitted notebooks evaluated against this actual test data. Upon submission, the test directory populates with approximately 200 recordings, each 10 minutes long.

The competition imposes specific requirements for code submissions, including submitting entries through notebooks, a maximum runtime of 120 minutes for CPU notebooks, disabling GPU notebook submissions and internet access, and permitting freely available external data, including pre-trained models. This research concentrates only on the 2023 data since Kaggle states that the training data has all relevant files, and no further benefit can be obtained by searching for more [31]. However, the project will deviate from specific rules about GPU usage and maximum runtime to more broadly explore the question of birdsong classification.

1.2.3 Audio Ratings

Data is also collected via PAM where recording devices are set up in specific locations to capture bird calls over time, allowing for data collection with minimal bird disturbance. In addition, a platform like Xenocanto.org leverages the power of 'citizen science', where bird enthusiasts worldwide contribute their recordings, leading to a large and diverse dataset with broad geographic and species coverage. As recordings are made in the wild, a particular concern for classification is noise, with recordings often capturing weather (such as winds and rain), human interference, and other wildlife. Limiting the effects of the noise will be an essential step in preparing the audio for classification.

XC encourages users to rate the quality of their audio uploads, giving them a gauge of the recording quality. However, details of the rating system are unclear, with no formal description of how ratings are assigned. The Xeno-Canto website suggests a classification from A to E based on the signal-to-noise ratio [21].

Signal-to-Noise Ratio (SNR) is a measure used to quantify how much noise has corrupted a signal. It represents the ratio of the power of a signal (meaningful information) to the power of background noise. In audio recordings, a high SNR means that the primary sound (like a bird call) is much louder than the background noise, making it more transparent and easier to identify. It is important to note that SNR can have values below 0dB, as the noise is measured over the entire audible frequency band, while the target call typically occupies a much narrower band. Measuring SNR can be problematic due to the need to estimate or model the noise in a signal [22].

1.3 Challenges

Audio data, mainly when sourced from natural environments like bird songs, exhibits high variability and is often contaminated with noise. This variability necessitates a robust preprocessing pipeline to enhance data quality and provide consistent input to classifiers. Bird song recordings, especially those sourced from platforms like XC or captured via PAM, present a variety of challenges:

- Diverse audio quality and length: PAM recordings can be long, often left running in the wild. In contrast, recordings contributed by birders might be captured in a glimpse on various handheld devices, leading to variability in quality and duration.
- Non-stationary noise: Recordings in both wild and urban settings frequently capture unwanted non-stationary ambient noise (whose characteristics change over time). Examples include traffic, weather, and human interference.
- Subjective quality ratings: Platforms like XC attempt to classify sounds based on SNR. However, this rating process remains subjective and can be unreliable.

The presence of noise in bird audio recordings can significantly impact classifier performance. Noise introduces variability into the training data, potentially leading to overfitting, a process where the model learns the noise rather than the bird call. Noise complicates feature extraction, which can obscure vital bird call features and require additional preprocessing and computational overheads. The Macaulay

Library provides guidelines for rating audio quality in bird recordings, stating the most important factor for rating audio quality is how loud the target bird sound is compared to the background noise [20].

Given the challenges posed by noise, preprocessing becomes paramount. Effective noise reduction techniques, such as spectral gating, enhance SNR and make bird calls more discernible. Segmenting the audio into individual units can help isolate bird calls from background noise. Adjusting sampling rates can aid in focusing on relevant frequency bands, further clarifying the bird calls and limiting the number of data points being processed for training.

Many published studies on bird song classification underscore the significance of manual inspection to ensure the quality of individual species samples [85, 12]. However, these studies often omit details on species selection, the quality of training samples and the complete preprocessing steps involved in preparing their training data. The variability in data quality, diverse recording equipment, and ambient noise underscore the importance of preprocessing in bird song audio classification. Effective preprocessing enhances data quality and ensures that classifiers are trained on consistent and representative data.

1.3.1 Deep Learning Challenges

Deep learning networks have specific challenges based on the chosen library (e.g., Keras, TensorFlow) and the implemented model. Issues related to GPU/CPU compatibility can arise, with errors due to memory allocation often occurring if too much computation is demanded. Challenges emerge if the shape of the data passed to the network is inconsistent with the network's expectations or if data types are inconsistent. For example, the network may expect real-valued numbers and will error if passed complex values, leading to potential program crashes.

The batch size can dictate how much memory the pipeline requires, and often, a batch too big will unexpectedly crash the system. The GPU environments themselves also have limitations on session times. Google COLAB will time out without enough user interaction; however, these issues may be avoided by paying for a subscription service, which grants access to more significant memory and processor resources [28].

Debugging problems in deep learning can be time-consuming due to numerical errors, training anomalies, and obscure error messages [88]. Careful debugging is required to trace issues like tensor shape mismatches, unsupported data types, and computational resource limitations.

Extracting wavelet features may present further challenges. The CWT offers a flexible, high-resolution method for signal analysis however, this flexibility can increase computational demands. CWT provides a shift-invariant representation (remains the same following a time shift), ideal for visualising frequency changes over time. However, the continuous nature of the CWT can be computationally intensive, producing vast amounts of coefficients. Choosing appropriate features like wavelet scales is crucial, impacting computational cost and classifier performance. Decisions around resolution, scales and coefficients must balance information gain versus computational needs, too many scales or coefficients may overload hardware and extend training times. Too few may not provide the models with enough feature information to enable successful identification.

Another primary challenge in deep learning is the need for large datasets ('big data') to train effectively. The sheer volume of data required can strain computational resources [66]. Deep models, especially when trained on big data, can have prolonged training times, requiring powerful GPUs for efficient training. Implementing and maintaining the code for deep learning pipelines, especially when integrating preprocessing, training, and evaluation, can be intricate. The performance of deep learning models is also sensitive to hyperparameters, the fine-tuning of which demands extensive computational resources and time.

CNNs have shown exceptional performance in computer vision tasks, with success relying heavily on big data to avoid overfitting. Overfitting occurs when a model learns to fit the training data perfectly but fails to generalise well to unseen data. Data augmentation is one common technique to mitigate the challenges posed by limited data that enhances training datasets by varying pitch and speed or introducing noise. This allows deep learning models to generalise to real-world variations [71].

In the context of birdsong classification, the challenge is further exacerbated. The British Trust for Ornithology's (BTO) Acoustic Pipeline offers state-of-the-art software to assist ecological research. Their species identification pipeline has processed over 500 terabytes of audio recordings from over 2000 users in 25 countries since its launch in 2021 [23]. While this might seem substantial, deep learning models, especially those designed for complex tasks like birdsong identification, require vast amounts of diverse data to achieve optimal performance. The diversity in bird species, their vocal variations, and the environmental noise in recordings make this task even more challenging.

1.3.2 Bird Song Complexity

Birdsong is a complex auditory signal that varies significantly across different species, populations, and even individual birds. The variability in these calls presents a unique challenge for learning classifiers aiming to identify and categorise them. Birds of the same species from different geographical locations may produce distinct vocalisations. For instance, the D-syllable of the chick-a-dee call of the black-capped chickadee (Poecile atricapilla) exhibits notable differences across the three types [5]. Even within a single population, individual birds may have unique call variations. A single bird's call can vary depending on the time of day, season, or age, and birds will sing at higher frequencies to thrive in urban settings [6]. Other studies have shown that members of the same blackbird species who live in the forest environment have higher testosterone levels than their city-dwelling counterparts, resulting in a deeper voice [49].

1.4 Problem Statement

The BirdCLEF competition has seen a variety of methodologies, with most entrants preferring the Fast Fourier Transform (FFT) and spectrograms [87]. Introducing wavelets to BirdCLEF could offer new avenues for performance improvement. Wavelet transforms, which provide a more flexible time-frequency representation, are often overlooked. This project explores a new method of processing bird songs using wavelets that offer improved time-frequency representations.

This research aims to contribute by developing a robust preprocessing and wavelet feature extraction pipeline to match or exceed published accuracy rates of over 85% using the training data provided. Findings could also benefit bird conservation by improving species identification and machine learning models in other audio-processing tasks.

The diversity of recording devices, the range of environments and the varying skill levels of authors in the raw data collection pose significant challenges for research. Not filtering samples retains the real-world data's natural variability and noise, testing whether an unmodified XC crowd-sourced dataset is sufficient or if extensive filtering and curation of training data are required for accurate species classification.

1.5 Aims, Objectives and Questions

1.5.1 Primary Motivation

This research aims to challenge and check the validity of published classification results using crowd-sourced, uncontrolled audio data via deep learning methodologies. To achieve this primary aim, the research is organised around the following objectives and research questions:

1.5.2 Preprocessing

- Objective: Investigate noise reduction and segmentation techniques to create enhanced song sample profiles (for example, reduced ambient background noise, less silence or noise between bird calls) without manual screening.
- Question: How effective are these techniques in enhancing the quality of the raw audio samples?
- Objective: Implement voice activity detection (VAD) and noise reduction algorithms to refine and enhance the quality of samples from raw recordings.
- Question: What is the impact of the VAD segmentation on the raw audio samples?

1.5.3 Model Development

- Objective: Create a TensorFlow pipeline using CWT wavelet features for CNN classification, managing heavy computational requirements.
- Question: How can a TensorFlow pipeline be optimised for CNN classification using CWT features? What are the batch sizes and wavelet scale limits for computation in a single GPU notebook?

- Objective: Train a CNN model on the 2023 BirdCLEF data without complex augmentation, evaluating preprocessing and quality of crowd-sourced data. Aim to exceed the benchmark 50% F1 score and compare with published results and a more traditional feature-based classifier.

- Question: Can a model be trained within a 12-hour timeframe? How does a CNN model train on un-augmented, balanced 2023 BirdClef training data in terms of accuracy and against other models?

1.5.4 Evaluation and Validation

- Objective: Establish methods to minimise overfitting and validate general performance on unseen test data.

- Question: What techniques effectively minimise overfitting when training a CNN on limited data? What impact do they have?

- Objective: Compare the performance and efficiency of a deep learning CNN model with a traditional classifier and published results such as those discussed in Chapter 2.

- Question: How does the wavelet feature CNN compare with a traditional classifier? Are published results achievable using random, uncurated sets of publicly crowd-sourced data?

1.5.5 Research Approach

The research approach is primarily quantitative, involving the application of audio processing techniques and evaluating the impact on classifier performance. The performance will be quantitatively measured using accuracy, precision, recall and F1 score. There will be minimal qualitative analysis of the effects of the preprocessing by visual inspection (for example, via waveforms and spectrograms) and listening to audio samples due to time constraints. However, it should be noted that no qualitative measures will be used for species or sample selection.

1.5.6 Limitations

In light of the computational requirements and challenges discussed in Section 1.3, this project will focus on classifying a subset of species from the BirdCLEF dataset. The subset of species with the most samples will be selected.

Given resource constraints, manual inspection and filtering of audio recordings was not feasible for this project. Instead, the unmodified BirdCLEF dataset from Kaggle is used as provided, without any preselection or quality assessment of samples. This allows the research to test the sufficiency of the raw crowd-sourced data for training accurate models without relying on manual curation, a unique limitation of this research. Rather than hand-picking high-quality samples, the aim is to evaluate whether uncontrolled contributions can still produce helpful results.

The research will investigate preprocessing methods for feature extraction using freely available and well-documented Python libraries like TensorFlow and sklearn [1, 56]. A random, balanced sample of data will be used for training and evaluation to align with computational resource limitations. This may uncover areas for further research echoing the plans of BirdNET developers to create an app that can classify newly captured data on limited resources without an internet connection [15].

Several constraints limit the scope:

- Species subset: Nine species are randomly selected based on similar studies, allowing for meaningful comparison to numbers in related work [34, 68].

- Model simplicity: A simplified CNN architecture with few layers is employed rather than complex pre-trained networks due to computational constraints. This may limit the complexity of the features the model can learn.

- Data constraints: No external datasets, heavy data augmentation processes, or pre-trained models are allowed.

- Compute limits: designed for a single CPU or GPU-enabled notebook environment.

- Metrics: Intuitive classification metrics like accuracy and F1 score are used rather than mAP for ease of understanding and comparison.

The focus is specifically on preparing a BirdCLEF subset, extracting wavelet features, and evaluating a straightforward CNN architecture. Further work could expand the species, leverage ensembles, and scale up computational resources.

1.6 Structure

The remainder of the thesis will be structured as follows: Chapter 2 provides an overview of current mobile applications demonstrating the effectiveness of birdsong classification algorithms and introduces the underlying concepts allowing them to work. Chapter 3 provides an overview of relevant literature on bird song classification, wavelet transforms, audio preprocessing, and frequency analysis to inform the full-process design. Chapter 4, Methodology, details the data preparation and preprocessing pipeline, model architecture and training methodology used in the experiments. Chapter 5 presents the experimental results from the different models and configurations and discusses the impact of preprocessing methods. Chapter 6 interprets the results, relates them to published study results, revisits aims and objectives, highlights limitations and identifies future work. Chapter 7, Conclusion, evaluates the overall achievements and reflects on the personal scholarly journey.

Chapter 2

Contextual Background

Chapter 2 provides an overview of current mobile applications for birdsong classification, introducing the fundamental Fourier and Wavelet transform processes and their associating visualisations, spectrograms and scalograms. The chapter concludes by proposing to train CNN classifiers using the coefficients of the CWT and highlights the advantages of this approach.

2.1 Classification Applications

In recent years, the study of birdsong has gained significant interest from the scientific community and bird enthusiasts ('birders') across the globe. Mobile applications (apps) can recognise bird songs from phone recordings, a skill that distinguishes amateur birders from professionals [8]. However, birdsongs are incredibly complex and diverse, with Lazuli Buntings using over 140 distinct syllables in their songs. This challenges computer algorithms, especially on mobile devices, as even advanced products struggle to match human ability in distinguishing between bird calls [68].

Several major universities, such as the Cornell Lab of Ornithology, and private companies, such as Wildlife Acoustics, have spun off commercial phone versions of their technology, which work in a manner very similar to the well-known song-identifying app Shazam [69]. The market for bird identification apps has increased, with users willing to pay subscription fees upwards of $9.99 per year for the most advanced products. Fierce competition has emerged, especially since the 2017 release of Song Sleuth, which recognises 200 common species using sophisticated algorithms [48].

Song Sleuth generates spectrograms (graphs displaying the frequency spectrum of bird audio recordings) and waveforms to aid the analysis of users' recordings. Identification is enhanced by geotagging each recording and refining suggestions based on location, and users can share recordings to get second opinions on unusual finds.

Figure 2.1: Merlin ID Spectrogram [53]

Merlin Sound ID from Cornell Labs employs CNNs to provide real-time species suggestions by comparing recordings against its database of 1,054 species [53]. Figure 2.1 shows a sample spectrogram from the Merlid ID app service. Users contribute new samples to improve the algorithm, uploading them to the

eBird checklist and tagging the species audible in the recording. Based on Cornell's earlier free BirdNet algorithm, the app generates a spectrogram allowing users to mark specific portions of the image ('annotations') corresponding to their bird chirp sample. As a result, this contributes to Merlin's database with precious hand-labelled segmented recordings, which can be challenging to obtain when attempting to capture high-quality bird song samples in the wild.

The app encourages anyone with a smartphone to make recordings, keeping a leaderboard of the highest contributors. However, the developers also mention the low quality of mobile phone recordings and explain how the accurately annotated data is needed as a compromise for successfully training the models. The website estimates that 'more than 100 recordings are needed per species' and that only 20% of birds have enough recordings to start training Merlin.

Despite all this, bird identification apps struggle to achieve consistently accurate results; for example, Song Sleuth often misclassifies the trill of the Chipping Sparrow due to ambient noise and overlapping songs [44]. Poor user reviews (averaging 2.4/5 stars) highlight the engineering difficulties of creating a reliable bird classification model [58].

In contrast, Cornell's Merlin application achieves higher user ratings for classification, benefitting from the manually annotated spectrograms submitted by experienced birders and ornithologists. Although inconsistencies remain, Merlin can struggle to differentiate similar species and occasionally misclassifies calls [57]. Whilst promising current applications have limitations requiring further refinement and evaluation. Expert ornithological knowledge remains vital to validate results.

2.2 Fourier Transform

All of the discussed applications use the Fourier Transform (FT) in their processes. An audio signal can be visualised in the time domain as a Waveform displaying amplitude over time. It can also be transformed into the frequency domain as a Frequency Spectrum showing power at each frequency component. While the waveform represents changes over time, the frequency spectrum shows the signal's constituent frequencies but lacks timing information.

The FT is a mathematical technique that decomposes a signal into its constituent sinusoidal waves, transforming signals from the time domain into their frequency components. However, while the FT can analyse any signal, it provides a global view of its frequency content without indicating when specific frequencies occur [10].

The FT transforms the signal to the frequency domain, revealing the frequencies present but losing time details. This works well for stationary audio but fails for signals where frequency content varies with time (non-stationary), like birdsong, and it is these limitations that led to the development of time-frequency techniques like the Short-Time Fourier Transform (STFT).

2.2.1 Short-Time Fourier Transform

The STFT addresses the lack of time localisation in the FT by computing the transform on short windows of the signal. The STFT splits the signal into segments and computes the FT separately on each window, capturing how the frequency content evolves [10].

The main parameters of the STFT are the window size and the hop size (the amount by which the window is shifted for each computation). The window size determines the trade-off between time resolution and frequency resolution: a larger window gives better frequency resolution but worse time resolution, and vice versa. The hop size determines the overlap between consecutive windows: a smaller hop size gives more overlap, which can result in a smoother spectrogram but also requires more computations. The choice of window size is critical; larger windows improve frequency resolution by sacrificing time resolution. This trade-off is an intrinsic feature of the STFT [60].

2.2.2 Spectrogram

The spectrogram is a visual representation of the STFT. It displays how signal frequency and power change over time, similar to having many frequency spectra stacked side by side, each representing a tiny time slice. Frequency is shown on the y-axis up to the maximum (Nyquist) frequency dictated by the sampling rate according to the Nyquist-Shannon sampling theorem. The Nyquist frequency is the maximum frequency that can be accurately captured during sampling without aliasing [43]. Aliasing occurs when a signal is undersampled and can cause high-frequency components to appear as lower-frequency components, distorting the signal when reconstructed. According to the Nyquist-Shannon

sampling theorem, a signal must be sampled at least twice as fast as its bandwidth to be captured accurately [43].

2.3 Wavelet Transform

This section will cover the development of Wavelet Theory and outline the process for calculating the coefficients (and, therefore, the scalogram). Further details on the choice of wavelet and the scale calculations are covered in Chapter 4.

The concept of wavelets can be traced back to the early 20th century with the work of Alfred Haar. In 1909, Haar introduced what is now known as the Haar wavelet, considered the most straightforward wavelet type [29]. However, the term "wavelet" was not used until the 1980s. The development of wavelet theory was motivated by the need for a tool to handle irregular patterns and signal trends, which the FT could not do effectively. The wavelet transforms gained significant attention in the 1980s, primarily due to the work of Jean Morlet, a geophysicist, and Alex Grossmann, a theoretical physicist. They developed a new family of mathematical functions known as Morlet wavelets, which are particularly useful for analysing seismic signals. Their work laid the foundation for the development of the continuous wavelet transform [47].

In 1986, Yves Meyer constructed the first orthogonal wavelet basis, a significant milestone in the history of wavelet theory [45]. This work was further developed by Stéphane Mallat and Meyer, leading to the development of the multiresolution analysis and the wavelet decomposition and reconstruction algorithm, widely used today [41]. In 1994, Ingrid Daubechies, a Belgian physicist and mathematician, published "Ten Lectures on Wavelets", which has since become a standard reference in wavelet theory, providing a comprehensive introduction to the theory and application of wavelets [16].

2.4 Continuous Wavelet Transform

The CWT provides a time-frequency representation using wavelets. Unlike the FT's sinusoidal basis functions, the CWT uses shifted and scaled wavelet functions as bases [64], allowing multi-resolution analysis, which provides higher resolution than Fourier techniques. This makes the CWT particularly useful for analysing non-stationary signals (like most real-world audio and birdsong), where the frequency content can change over time [2].

The CWT is computed by correlating the signal $x(t)$ with a wavelet function $\psi(t)$ at different scales s and positions u:

$$CWT_x(s,u) = \int x(t)\psi_{s,u}(t)dt \tag{2.1}$$

where

$$\psi_{s,u}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{2.2}$$

Here, $\psi(t)$ is the mother wavelet, s is the scale parameter, and u is the position parameter. The mother wavelet $\psi(t)$ is the base wavelet function that is scaled and shifted to perform the transform. Some common examples are the Morlet, Gaussian and Haar wavelet. For instance, the Morlet wavelet, as shown on the left of Figure 2.2, is defined as:

$$\psi(t) = ce^{i\omega t}e^{-\frac{t^2}{2}} \tag{2.3}$$

Where c is a normalisation constant and ω is the central frequency. The mother wavelet provides the basic waveform shape and properties on which the CWT analysis is built.

Scale s is inversely related to frequency, with higher scales corresponding to lower frequencies and vice versa. When the scale is increased, the wavelet function becomes stretched out, prolonging its period and allowing it to match low-frequency components in the signal. The wavelet function is compressed at lower scales, shortening its periodicity to match high-frequency components. Hence, low scales focus on high frequencies, while high scales focus on low frequencies.

Generally, the relationship between scale, frequency, and sampling rate is given by:

$$f = \frac{\text{sampling rate}}{\text{scale}}$$

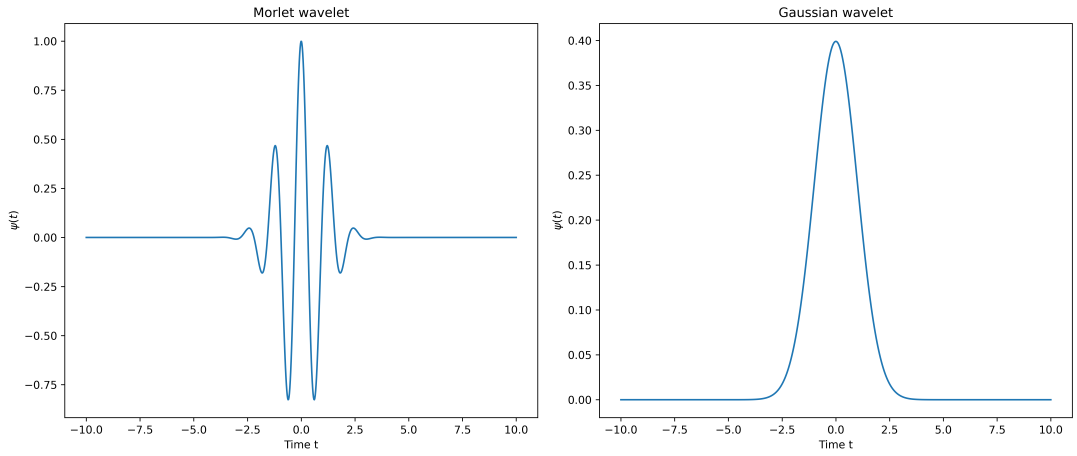

Figure 2.2: Morlet and Gaussian Wavelet

.

For example, for a 32 kHz sampled signal wanting to analyse down to 20 Hz (lower human hearing limit), the scale would be:

$$\frac{32000}{20} = 1600$$

.

Note that this relationship can be specific to the wavelet used in the CWT, and a scale step size can be used to determine frequency resolution. A smaller step offers higher resolution but increases computation time as more scales must be computed.

The output of the CWT is a set of coefficients that tell you how much of the signal is represented by each basis function. The coefficients $CWT_x(s, u)$ indicate how closely the wavelet matches the signal at each scale and position. The CWT is advantageous for non-stationary signals like birdsong, providing good time resolution for transients and frequency resolution for stable components [42]. Visualising the CWT coefficients generates the scalogram.

2.4.1 Scalogram

The CWT scalogram is a time-frequency representation that uses wavelets to analyse the signal. Scalograms offer variable resolution, whereas spectrograms have a fixed resolution across frequencies. The STFT derives the spectrogram by breaking down the signal into sinusoids of different frequencies. In contrast, wavelet analysis derives the scalogram by breaking down the signal using wavelets (small wave-like oscillations).

The main advantage of wavelet analysis over FT is variable resolution, providing finer frequency resolution at lower frequencies and finer time resolution at higher frequencies, making them especially useful for analysing signals with time-varying characteristics, like bird songs.

Figure 2.3 shows the Waveform, Frequency Spectrum, Spectrogram, and Wavelet Scalogram of an audio clip from the 2023 BirdClef dataset featuring Egyptian Goose (Alopochen aegyptiaca) with a sampling rate of 22050 Hz. The spectrogram for the Goose's call shows brightness across a vast range of pitches, suggesting activity at nearly all frequencies. In contrast, the scalogram offers a more precise picture by highlighting specific areas, giving a more accurate understanding of where the most significant changes in the call occurred.

The spectrogram provides an overview of the call's features, while the scalogram highlights particular areas. This research exploits this unique advantage by training the deep learning model on CWT coefficients, analogous to training the CNN model on images.

2.4.2 Advantages of CWT for Birdsong

The CWT provides several advantages for analysing birdsong and other non-stationary signals relative to Fourier-based techniques like the STFT. The CWT offers a principled solution to selecting an appropriate window size well-suited for time-frequency analysis and has been successfully applied in previous studies

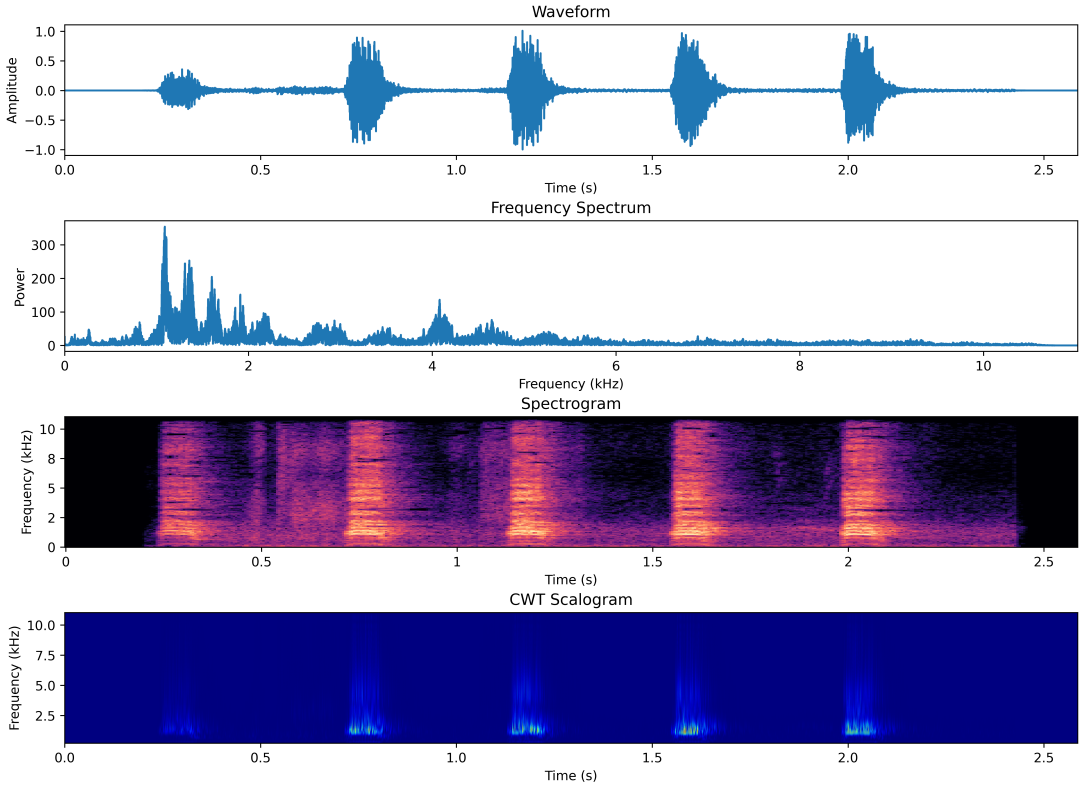

Figure 2.3: Egyptian Goose Audio Visualisation

[34]. The wavelet function is a fully scalable modulated window, enabling multi-resolution time-frequency analysis [16], providing accurate timing for capturing sudden changes and precise frequency measurement for stable components - suiting birdsong's mix of fast and slow elements [70].

However, the CWT has a higher computational cost than the STFT since the CWT is computed at several scales, while the STFT uses a fixed window size. Balancing accuracy and efficiency can be achieved during model training by incrementally increasing the scale while considering computational resources, where reducing the scales will accelerate CWT computation but sacrifice resolution. Compared to the STFT, the CWT provides a more accurate and higher resolution time-frequency representation, but it also has redundancy (many coefficients close to zero) that may be minimised via soft thresholding technique [17].

This research investigates an approach for classifying birdsongs using CWT coefficient-based CNNs (CWT-CNN). Instead of using spectrogram or scalogram images, the CWT coefficients are used directly as input, allowing them to be analysed and amended. Thresholding the CWT coefficients can reduce noise and computational requirements while retaining helpful information [17] making it ideal for analysing non-stationary noisy birdsong signals.

The next chapter reviews the use of CNN architectures and the CWT in related works.

Chapter 3

Related Work

This section will review the use of CNN architectures in the published literature, exploring examples of related works in-depth and supporting studies to influence the model and feature extraction decision-making explored in Chapter 3.

The use of artificial neural networks for bioacoustic species detection and classification dates back over two decades. In the early 2000s, neural network classifiers were used to differentiate sounds from grasshoppers [14] and bat species [54]. By 2014, the shift to deep learning led to significant performance improvements, as demonstrated by the BirdClef competition cMAP scores discussed in Section 1.2. Turker et al. [82] used a new handcrafted features-based machine learning model using the wavelet transform to classify bird sounds with 96.67% accuracy on an 18-class dataset. Toth and Czeba [81] used a CNN to classify bird species based on spectrograms, achieving a MAP score of 40% for the main species.

Wavelets have been successful in various fields, such as medical imaging [24], time-series analysis [63], biomedical engineering, and finance. The literature highlights many advantages of transitioning from Fourier to wavelet bases, such as superior compression over JPEG [84]. In biomedical engineering, wavelets enable more precise ECG analysis for detecting heart abnormalities [40], while in finance, they are used for stock prediction, showcasing suitability for non-stationary volatile time series [90].

Prior work has demonstrated successful wavelet-based analysis for mosquito detection [34] using higher quality but minimal duration audio data. This project investigates whether similar techniques could be applied to longer-duration, lower-quality crowd-sourced recordings. Kiskin et al. [34] used wavelet spectrograms as inputs for CNNs to detect mosquitoes. The authors use a CNN architecture that classifies raw audio data wavelet representations, treating the task as an image classification problem and using the wavelet scalogram as input. The authors found that their CWT-CNN outperforms traditional classifiers with F1 scores above 0.925. This presents an opportunity to explore if similar success can be achieved on the BirdClef 2023 dataset.

Due to limited data, the authors chose a simple CNN architecture with a single convolution layer, and while promising, this technique warrants further exploration to exploit its potential advantages fully. More complex networks could enhance performance by extracting intricate features and improving the model's generalisation across different bird songs. With more layers, CNNs can learn and abstract various representations from the wavelet coefficients, which are effective in large-scale video classification [32].

The authors justify their choice of the CWT over the more commonly used STFT, citing superior time-frequency resolution and selecting the 'bump' wavelet family for its popularity in time-frequency analysis. However, they do not analyse raw data characteristics to match wavelet properties, which has no transparent methodology as suggested in [68].

For comparison, the authors benchmark several conventional classifiers, including random forests, with explicit features designed to encode the raw data. They extract ten standard acoustic features from the raw data, such as spectrogram slices and various entropy measures [25]. However, relying on hand-selected features may limit the model's adaptability to new datasets or tasks.

Given the study's use of controlled, short-duration samples, it is uncertain whether mosquito sound analysis methods can accurately analyse bird vocalisation data from Kaggle. This research proposal is more challenging than the research where the authors required low sampling rates (8kHz) for their audio data, which was also restricted to a frequency range of 150-750Hz.

Selin et al. [68] classify eight bird songs using features derived from wavelet analysis and neural

networks. The authors conducted a study on bird species recognition, using 3132 sounds for training and testing purposes. The results showed 78% accuracy for select sounds like mallard and greylag goose. The authors justify using the wavelet transform, citing its ability to provide frequency and temporal resolution while analysing transients and spikes. After initial model testing, the Debauchies wavelet produced optimal decomposition. However, they note that reliable algorithms do not yet exist for automatically selecting suitable wavelets and decomposition levels.

To extract shift-invariant features, the authors reduce the wavelet coefficients to four normalised parameters: maximum energy, position, spread, and width. This feature vector is used for classification training. They note that while beneficial for learning and generalisation, neural networks have limitations like fixed output classes requiring retraining with the addition of new species. The authors emphasise the importance of segmentation and noise reduction in obtaining high-quality, comparable samples, manually checking all sounds after post-processing, and discarding sounds recorded in noisy environments or with inseparable groups of birds. They also limited their selections to one type of call per species, reducing variance in the training data.

Other recent work highlights the continued reliance on manual processing and hand-picking of bird audio data. One study acknowledges the challenge of working with noisy samples, stating they cannot process low SNR signals soiled by weather conditions [13].

Zhang et al. [87] propose combining CNNs trained on various spectrogram representations to enhance bird species identification, arguing that spectrograms provide superior time-frequency resolution for extracting features from the audio. Their models are evaluated on the BirdCLEF2019 dataset of 350 hours of recordings covering 659 species, achieving a 0.135 cMAP score, slightly below the top score of 0.140-0.160. The authors propose using deep CNNs to learn complex features and suggest careful parameter tuning with sufficient samples of all bird species for best performance. They also caution that downloaded data may hinder model performance, and thorough data selection is needed to avoid overfitting.

A study on denoising earthquake seismogram signals using the CWT is discussed in [46]. The time-domain seismogram is transformed into a CWT scalogram, allowing for analysis in both the time and frequency domains. The Morlet wavelet is chosen for its Fourier-like properties, which help to reduce the effect of seismogram interruptions. 'Soft thresholding', a technique that sets small CWT coefficients to zero to remove noise, is applied to the scalogram, suggesting a potential application to audio captured in noisy natural environments.

These studies show that deep learning approaches are effective for bird species identification. However, there is heavy reliance on manual sample selection and preprocessing like denoising and segmentation. Opportunities remain to investigate whether such labour-intensive cleansing is essential for success on uncontrolled public datasets.

Although the CWT-CNN detection for mosquitoes shows promise, it must be validated using real-world bird data. Short, controlled audio samples differ from extensive, noisy, crowd-sourced collections and extending this research to uncontrolled conditions would better demonstrate effectiveness.

Overall, the literature establishes deep learning networks and wavelet transforms as promising techniques for bird species classification. However, some authors acknowledge that accuracy still falls short of human experts [68]. Opportunities remain to study the performance of wavelet transforms and validate performance with public datasets. Without expert feature engineering, CNNs can extract important hierarchical features from raw audio data, and this study aims to train classifiers using CWT techniques instead of relying on feature crafting. A traditional classifier will be prepared using the feature-based methods discussed above to provide a performance benchmark.

The following chapter covers how bird species are classified from raw audio data, including data selection, audio preprocessing and model training. The model architecture and design, as well as training methods, are also discussed.

Chapter 4

Methods

This Chapter outlines the approach to classifying bird species from raw audio data. It discusses the steps taken from data selection to model training. The "Data Analysis" section examines the dataset statistics and sampling considerations. "Audio Preprocessing" describes techniques like noise reduction and segmentation applied to refine the raw audio data, underscoring the significance and complexity of preprocessing for audio classification. "Wavelet Transform Coefficients" describes using the CWT for time-frequency feature extraction and postprocessing noise removal. "Model Architecture" outlines the CNN model and a traditional classifier that contrasts performance. Finally, "Model Training" describes the model optimisation process, the use of GPU acceleration, and techniques employed to minimise the effects of overfitting.

4.1 Methodology

This research presents a CWT-CNN for classifying bird species using noisy audio recordings. The model of the whole recognition process is displayed in Figure 4.1.

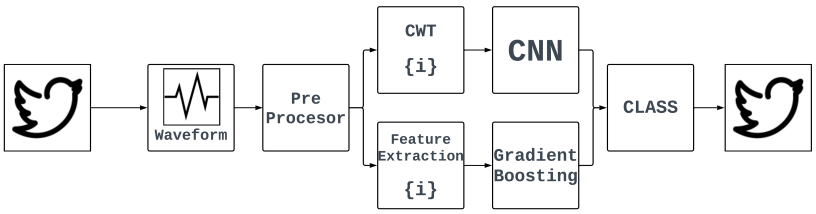

Figure 4.1: Modelling Process Flowchart

Order of preprocessing is crucial. The audio samples underwent several preprocessing steps such as removal of complete silence intervals, noise reduction using a non-stationary filter [51], removal of 'non-speech' periods using a voice activity detection (VAD) algorithm [26], and segmentation of soundtracks into equally-sized chunks to effectively train the CNN. The effects of the preprocessing pipeline were manually tested using visualisations and some listening, purely to establish a working process. It should be noted that no recordings were selected on perceived quality, and the species were chosen solely based on the maximum number of recordings with a suitable duration, ensuring no selection bias creeps in.

Two models are trained, with one discussed in detail. The CWT-CNN is the primary objective of this research to test the application of CWT on longer duration, random bird sounds and establish a working preprocessing pipeline without hand-selecting training samples. The traditional classifier has been developed using published methods [68, 87, 72] demonstrating the effectiveness of the more common STFT approach and manual feature extraction.

A 'bottom-up' management technique has been employed throughout the pipeline and model development process. The bottom-up approach concentrates on building up smaller, manageable tasks that

all contribute to one big goal [79]. In this research, this one big goal is to achieve classification results of over 50% whilst also developing an entire process pipeline.

An advantage of the bottom-up approach is the flexibility if changes or issues arise, meaning it works well if constructed iteratively. The audio preprocessing channels and models have been built up from essential elements, which are added to following successful testing and feedback. This iterative process is signified by the version control in the preprocessor and additional trials in the model training, testing and development cycle. The AudioPreproccesor version control log demonstrates the iterative development and is shown in Appendix B. Features are added incrementally, tested for effectiveness on small samples, and progressed on feedback from best-looking results. Table 5.1 in Chapter 5 lists the main iterative steps in the CNN model build and testing.

4.2 Data Analysis

This section outlines the dataset for bird species recognition and the sampling strategy employed. Only recordings from the 2023 Kaggle training dataset with a single primary species label were considered to focus on identifying individual species. Recordings containing secondary labels were excluded to analyse the impact of preprocessing and wavelet feature extraction without confounding factors (14636 recordings total). Other metadata fields, such as location coordinates and call type, were excluded from the analysis. The 333 unique call-type entries lacked information on classification methodology, so they were removed to align with the research aims of unbiased data processing. Of the total 3781 files, 471 had no labelled call type.

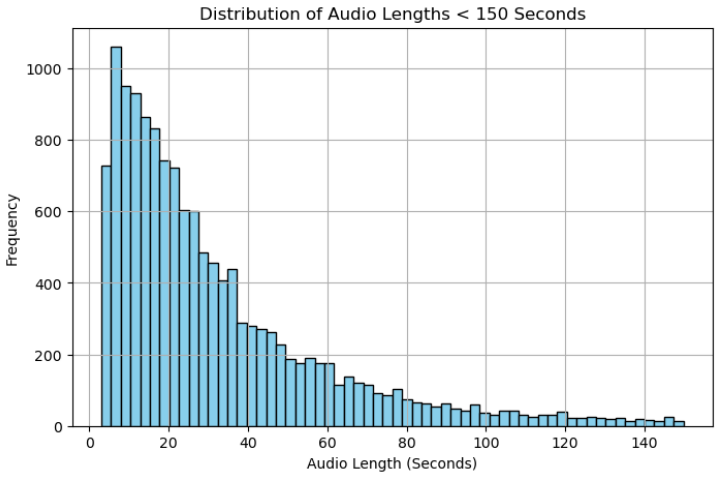

Figure 4.2: Audio Length Distribution Chart

Examining the distribution of audio lengths showed that 95% of recordings were between 3 and 150 seconds long (13850 files). Nine species were selected from this subset with maximum total audio lengths aiming to generate the most equal-length segments after preprocessing, thereby limiting imbalance effects on classifier performance. The sample size aligns with precedents using 8-10 species [34, 68].

Table 4.1 summarises the recordings per species, total length, and length statistics. Appendix A displays images of each of the nine species. Appendix A displays images of each of the nine species.

Using a balanced dataset, this research aims to match published results by randomly sampling the provided training data without relying on data augmentation or external additional datasets. Due to processing times being unknown at first, training and testing started with smaller sample numbers and increased with each development iteration. At all stages, the number of audio segments for each species was kept constant to eliminate the effects of imbalance.

4.3 Audio Preprocessing

This research aims to develop a robust pipeline for preprocessing bird song audio files to enable the training of machine learning classifiers and enhance the audio data quality without manual inspection. The importance of preprocessing phases like noise removal, signal decomposition, smoothing, and filtering

Table 4.1: Nine Species for Classification

Class	Common Name	Files	Total (s)	Mean (s)	Std (s)	Min (s)	Max (s)
0	Barn Swallow	476	15579	33	29	5	145
1	Common Buzzard	441	17050	39	33	4	150
2	Common House-Martin	379	12565	33	31	3	149
3	Common Sandpiper	484	11854	24	24	5	146
4	Eurasian Hoopoe	366	14030	38	30	4	150
5	European Bee-eater	374	11975	32	28	3	149
6	Thrush Nightingale	356	21619	61	35	5	148
7	Western Yellow Wagtail	472	14553	31	28	5	142
8	Willow Warbler	433	19657	45	33	5	148

is highlighted in recent studies [83], emphasising the impact of noise and irrelevant data, specifically when training a CNN.

The pipeline involves implementing a series of audio preprocessing techniques using Python and libraries such as librosa, scipy, and noisereduce, including steps for downsampling, reducing noise, applying a bandpass filter, removing silence, and segmenting. The success of each preprocessing step is evaluated through a series of experiments and confirmed via random manual sampling – using waveform, spectrograms and listening to the audio for impact – to assess whether the steps are working correctly. Resource limitations have restricted the tuning of these steps.

Due to the project limitations, not all audio can be inspected, as often does in the published studies. However, there may be scope to attribute the success of the preprocessing to the accuracy of classification. Ultimately, the impact of the pipeline will be evaluated via classification results.

The order of operations is designed to incrementally refine the audio by removing irrelevant data, retaining the valuable frequency band, increasing call density, segmenting, and downsampling.

Silence detection first minimises the impact on the moving noise average calculation, and then the noisereduce function [65] is applied to remove ambient noise in the recordings. Next, the remove silence function with VAD technology separates 'speech' (bird song) and 'non-speech' (background noise), aiming to improve the density of bird calls in each segment by removing 'non-speech'. A bandpass filter retains only the target bird frequency range, informed by analysis of typical bird vocalisation frequencies. The audio is then segmented into fixed 5-second chunks and downsampled to reduce the number of CWT coefficients for efficiently training the CNN.

4.3.1 silence detection

This first preprocessing step removes silent segments from the audio waveform using the bespoke detect_silences_waveform function. Initial visual manual inspection revealed blank parts in the raw recordings. Removing these upfront reduces the amount of data processed and allows more accurate noise reduction by avoiding misleading noise floor estimates.

The audio is divided into 0.25-second frames, and each frame's root-mean-square (RMS) energy is calculated. RMS energy measures the audio signal's average power; a high RMS energy may indicate a loud and powerful birdsong [78]. In contrast, low RMS may mean a quieter song or one more subdued by noise. A period of silence, therefore, will have rms energy close to or at zero. Frames with energy below a certain threshold are considered silent, and consecutive silent frames are combined into quiet periods, which are removed. The energy threshold is set to 0.01 to remove only completely silent segments.

Silence detection is performed first to avoid misleading subsequent noise reduction algorithms, which assume that low-energy areas contain 'non-speech' or, in this instance, the background sounds recorded between bird calls. After noise reduction, a bandpass filter retains only the target bird vocalisation frequency range, reducing data requirements.

4.3.2 noise reduction

The noisereduce library [51] offers stationary and non-stationary noise reduction. Stationary mode requires a sample noise clip to calculate a spectrogram threshold for gating frequencies. Non-stationary reduction allows the threshold to vary over time, more suitable for brief bird calls [18] amidst background noise.

The reduce_noise function applies non-stationary reduction using the sample rate, noise reduction proportion, and a 0.5s time constant for bird calls [18]. The prop_decrease controls attenuation percentage; a value of 1.0 gives 100%. Higher values risk distorting samples [19]. A prop_decrease of 1.0 was chosen to reduce noise while minimising distortion.

4.3.3 remove silence

The silence_removal function from audioSegmentation.py [26] extracts individual audio events from a recording by removing silent areas. It uses a semi-supervised approach with a classification model trained on high and low energy frames and applies dynamic thresholding to detect active segments. Consecutive silent low-RMS energy [78] frames are removed to isolate bird calls and increase call density in each segment. This clarifies the bird song from background noise, allowing more distinct features to be extracted from smaller windows where the aim is to condense the vocal signals and remove irrelevant noise between calls. The effects of these preprocessing steps are further demonstrated in Chapter 5.

Data: Audio Files
Result: Preprocessed Audio Files
for *each audio file* **do**
  Load audio file
  if *silence detection* **then**
    Detect and remove silent periods
  else
  end
  if *noise reduction* **then**
    Apply non-stationary reduce_noise
  else
  end
  if *remove silence* **then**
    Using VAD to separate 'speech'
  else
  end
  if *band pass filter* **then**
    Apply bandpass filter
  else
  end
  Save audio - downsample and segment
end

Algorithm 4.1: Audiopreprocessor Pipeline (AP)

4.3.4 band pass filter

Band-pass filters allow specific frequency components to pass while blocking others [75]. Urban noise contains lower frequencies with a 231 Hz peak in one study [6], while bird songs vary from 1300Hz to 7800Hz across studies [70, 18].

Following the bottom-up development approach, a 250Hz high-pass filter was applied to reduce low-frequency noise and retain bird vocalisations [18]. Limited time and resources led to the use of published frequency ranges. A detailed analysis of the training data could improve filter selection. Band-pass filtering isolates bird calls by removing non-vocalisation frequencies. The filter cutoff was refined based on literature ranges, but data-driven optimisation could be explored.

4.3.5 save audio

The BirdCLEF competition users employ five to ten-second segments for classifier training and the competition rules stipulate five-second windows for evaluation [3, 52]. This research uses five-second chunks at 12kHz downsampling, giving 60000 coefficients per sample, to match the standard approaches and competition constraints. This significantly increased the data and audio processing requirements

compared with similar CWT-CNN studies focusing on brief events like mosquito wingbeats, with audio typically under 200ms long and sampled at 8kHz [34].

The fixed five-second length ensures consistent model input, with padding for short samples. CNN training requires equal-sized arrays [86]. Audio outside 5s is trimmed, and padding is applied to shorter audio to conform all samples.

4.4 Wavelet Transform Coefficients

The CWT extracts time-frequency features from the audio signals, representing a signal in both time and frequency. This makes it well-suited for non-stationary signals like bird vocalisations, where frequency content varies over time [42]. Choosing appropriate wavelet scales is key for capturing distinguishing frequency patterns.

4.4.1 Wavelet Family and Scale

The Morlet wavelet is selected for its Fourier-like properties that enable frequency spectra analysis. Mohammed et al. [46] show the Morlet's effectiveness on seismic data, using soft thresholding of CWT coefficients to denoise signals and has been adopted for use on the bird audio collected in noisy environments. The Morlet wavelet's sinusoidal shape provides frequency resolution while maintaining localisation in time.

The PyWavelets' pywt.cwt function[80] computes the CWT using the signal, wavelet scales, and wavelet type. Downsampled audio is analysed at 64 linear-spaced scales from 200Hz to 6000Hz, estimated to cover vocalisation frequencies up to the Nyquist Frequency.

CWT coefficients are normalised by dividing by the maximum absolute value, an essential preprocessing step for the CNN [86], and avoids significant variances that could complicate learning. The resulting coefficients capture multi-resolution time-frequency patterns for classification training.

4.4.2 Denoising Wavelets

Background noises like wind or rain have varied frequency content depending on characteristics, for example, wind noise can contain low and high frequencies from gusts and raindrops. The preprocessing pipeline applied cautious noise reduction but could not eliminate all noise from every sample. Wavelet thresholding is another potential method for post-processing denoising.

Thresholding sets small wavelet coefficients, deemed as noise, to zero while retaining more significant, signal-related coefficients [17]. This assumes the bird vocalisation has the most important values where the wavelet matches its time-frequency patterns. Setting coefficients below a threshold to zero may remove most noise components.

However, thresholding risks also removing the actual signal if noise shares similar frequencies. Appendix C shows a raw goose call scalogram compared to thresholding at 5% of the maximum coefficient. This drastically reduces coefficients but retains the main vocalisation features. More adaptive thresholding methods may better isolate signals from noisy backgrounds [89].

4.5 Model Architecture

This section discussed the structure of the CNN used for bird species classification, outlining the specific layers and their roles in feature extraction and pattern recognition and the bespoke data generator function which ensures the model receives features efficiently for training.

4.5.1 CNN Overview

CNNs originate from biological processes observed in the human visual cortex, where neurons respond to restricted visible regions, being mimicked by convolutional layers. CNNs learn spatial features hierarchically, progressing from simple edges to complete shapes and objects, just like the brain's visual processing hierarchy [86].

Lower convolutional layers may detect edges, mid-layers combine these into shapes, and higher layers identify complex structures by compiling prior knowledge. Convolutional layers consist of filters (kernels) that scroll input data to learn features and share weights, reducing parameters for efficient learning [37].

Pooling layers downsample the feature maps, filtering less critical details like the brain does [86]. Fully connected layers at the end perform classification based on extracted features.

It is worth noting that CNNs are highly effective for tasks involving images, provided that they have access to extensive labelled datasets and sufficient computational resources. To avoid overfitting, which is often observed in deeper networks, regularisation techniques such as dropout are commonly employed [73].

4.5.2 Model configurations

The final CNN model comprised two convolutional and one dense layer and was developed in TensorFlow using a data generator for efficient batch processing. The tensor input shape was calculated from the CWT coefficient array produced by the data generator. Increasing the number of wavlet scales increased the size of this array accordingly. Dropout was applied for regularisation. Flattened features feed into dense layers, with a ReLU-activated 128-node layer and 9-node softmax output for the final classification of the nine species.

A complete list of CNN model parameters and variables is shown in Appendix D for future reproduction.

4.5.3 Data generator function

A Python generator function supplies training batches (BA) to the model on the fly for memory efficiency, looping through the data at each epoch (EP), yielding CWT coefficients and labels. The generator reads in audio files, computes CWTs using the configured wavelet and scales, and assembles fixed-size batches. On-demand batch generation reduces memory requirements compared to loading the entire dataset. Normalisation and thresholding are applied to the CWT coefficients for robust learning. The generator output shape matches the model input shape for seamless training.

4.6 Model Training

Training a neural network involves finding the kernels and weights that minimise the difference between predictions and ground truth labels. This is done using the backpropagation algorithm and an optimisation algorithm like gradient descent. Learnable parameters are updated based on the loss value [86].

The CNN model was trained on CWT coefficients from the audio recordings. Optimisation used Adam with a cosine annealing learning rate schedule for 20 epochs (EP). Sparse categorical cross-entropy loss was minimised, using GPU resources for efficient batch processing. The model was evaluated at each epoch on the validation set to track accuracy and loss—this guided architecture refinements like altering dropout probability rates.

4.6.1 Dropout

Dropout is a computationally cheap and effective way to improve generalisation error in deep neural networks by randomly dropping out nodes during training. This encourages each node to learn robust features useful across architectures, improving generalisation and reducing overfitting and works more effectively when there is a limited amount of training data [9].

The CNN applies dropout after the convolutional and dense layers; it is not used on the output layer. The default interpretation of the dropout hyperparameter is the probability of training a given node in a layer, where 1.0 means no dropout, and 0.0 means no outputs from the layer. Higher dropout rates can lead to node independence, but excessive dropout can cause underfitting [73]. To prevent this, iterative tuning guided by validation is necessary due to the limited training data. A higher dropout rate of 0.4 was tested during the development cycle.

4.6.2 Challenges and insights from the training process

To prevent the COLAB session from crashing, smaller batches were needed due to limited memory. Errors caused by inconsistent input sizes from various audio lengths were resolved through padding and trimming in the preprocessing pipeline — local COLAB storage optimised file transfer over the internet to minimise transfer times and free up storage space. However, when memory was freed after execution, the training

history was initially accidentally lost. Overall, hyperparameter tuning was constrained by long training times, but the iterative process still produced a robust CNN architecture for bird call classification.

4.7 Traditional Classifer

An XGBoost (XGB) gradient boosting model was trained using standard feature extraction methods for comparison. XGB is a popular gradient-boosting algorithm known for efficiency, scalability, and performance on complex data [33]. Gradient boosting combines weak models, like decision trees, into a robust ensemble by iteratively correcting errors [67].

Features included wavelet decomposition, STFT, and MFCCs as defined in [68, 87].

The XGB hyperparameters were optimised via grid search for regularisation. The iterative development process balanced model complexity and overfitting, as shown in Chapter 4, Table E.1. Comparing the CNN to a mainstream classifier like XGB provides a benchmark for the CWT approach. Appendix E shows a complete list of the XGB model parameters.

4.7.1 Metrics

Several metrics evaluated the multi-class bird species classifiers:

- Accuracy: Proportion of correctly classified samples.
- Precision: Ratio of true to predicted positives, important when false positives are costly.
- Recall: Ratio of true positives to actual positives is crucial when false negatives are detrimental.
- F1 Score: Harmonic mean of precision and recall.

The F1 Score is a robust measure of performance that balances precision and recall to provide a single metric for each class and when weighted across all classes [56, 1].

4.7.2 Computational Resource

The CNN model was trained in the Google Colab environment [28], utilising a Python 3 runtime on a Google Compute Engine equipped with a Tensor Processing Unit (TPU). The system had 35 GB of RAM and a disk capacity of 225 GB. The dataset was partitioned into a 70/15/15% split for training, validation, and testing purposes.

The XGBoost model was trained on a laptop machine running Windows 10 with a 64-bit architecture, Intel64 CPU, with a total of 15 GB RAM. The dataset was partitioned into an 80/20 split for training and testing. Additionally, 20% of the training data was set aside for validating the gradient-boosting model.

The next chapter presents the classification results and demonstrates the impact of the preprocessing pipeline steps.

Chapter 5

Results

This chapter presents the key findings and results of two model development processes. The classification results are reported for both the CNN and Gradient Boost (XGB) models, showing the impact of different architecture and parameter choices. The iterative development process is outlined to provide details of the experiments conducted. The data preprocessing steps of noise removal and VAD segmentation are presented visually, highlighting their importance in preparing the raw audio for analysis.

5.1 Development Cycle

Model development proceeded in an iterative fashion, using feedback from each trial to guide parameter adjustment and model refinement. Table 5.1 summarises the CNN experiments, showing the progression in accuracy as architectural (C = Convolution Layers, D = Dense Layers), sample size (n), wavelet type (WAVE) and scale (S) changes were made.

Similarly, the key milestones in the XGB pipeline are outlined in Appendix E, highlighting the boosts obtained from increasing sample size, adding transform extracted features and refining STFT window parameters. This process required careful interpretation of the features defined in the studies outlined in Chapter 3.

Table 5.1: CWT-CNN Iterative Development Cycle

TL	AP	n	WAVE	S	TH	C	D	DP	BA	EP	TRA	VAL	V_F1	TST	T_F1	t (s)
2b	v5	800	morl	14	-	2	2	0.25	16	3	0.49	0.46	-	-	-	4049
2b	v6	800	morl	24	-	1	1	0.25	2	2	0.45	0.40	0.30	-	-	3261
2b	v7	800	morl	24	-	3	2	0.25	8	5	0.60	0.40	0.35	0.38	0.30	5617
3	v8	540	morl	24	-	3	2	0.25	8	2	0.43	0.46	0.25	0.47	0.25	1670
4	v8	1000	cmor1-1	32	-	3	2	0.40	10	6	0.71	0.58	0.48	-	-	14121
5	v9	1000	cmor1-1	32	-	3	2	0.35	10	4	0.52	0.45	0.30	0.43	0.29	23009
6	v9	1000	Gaus2	32	0.05	3	2	0.35	10	3	0.57	0.58	0.47	0.57	0.46	20556
7	v9	1000	morl	64	0.05	3	2	0.40	15	6	0.77	0.58	0.55	-	-	40355
8	v9	1000	morl	64	0.05	2	1	0.45	10	3	0.74	0.57	0.51	0.57	0.52	38302

5.2 CNN Classification Results

The CNN model achieved reasonable F1 scores of 57% on the unseen test data, indicating that the CNN was able to learn distinct features and patterns in the CWT coefficients. These results validate the CNN as a promising new approach for bird call classification.

Figure 5.1 shows the final results on the test set for the best-performing CNN architecture. Combining convolutional and dense layers and regularisation through dropout produced improvements in classification.

Classification Report

	precision	recall	f1-score	support
Class_0	0.46	0.20	0.28	150
Class_1	0.44	0.27	0.34	150
Class_2	0.75	0.63	0.68	150
Class_3	0.75	0.67	0.71	150
Class_4	0.39	0.75	0.51	150
Class_5	0.72	0.76	0.74	150
Class_6	0.89	0.50	0.64	150
Class_7	0.48	0.61	0.54	150
Class_8	0.53	0.75	0.62	150
accuracy			0.57	1350
macro avg	0.60	0.57	0.56	1350
weighted avg	0.60	0.57	0.56	1350

Confusion Matrix

```
[[ 30  19   7   5  35   0   1  10  43]
 [ 21  41   0   5  47   2   1  21  12]
 [  0   1  94   1  10  12   2  15  15]
 [  1   6   1 100  31   0   0   5   6]
 [  4   9   0  11 112   1   0   8   5]
 [  0   2   2   1  13 114   1  16   1]
 [  5  11   4   4  23   4  75  14  10]
 [  1   4   7   0  11  26   2  91   8]
 [  3   1  10   6   8   0   2   8 112]]
```

Figure 5.1: CWT-CNN Final Trial Results

5.2.1 Overfitting and Dropout

Figure 5.2 highlights the problem of overfitting encountered during the training of the CNN and the effectiveness of dropout in regularising the network. The left-hand plot (relating to trial 2) shows the validation loss increasing immediately after one epoch. Following the addition of dropout by Trial 6, the loss decreases in line with the training loss before finally showing signs of overfitting after the third epoch, as shown in the right-hand plot.

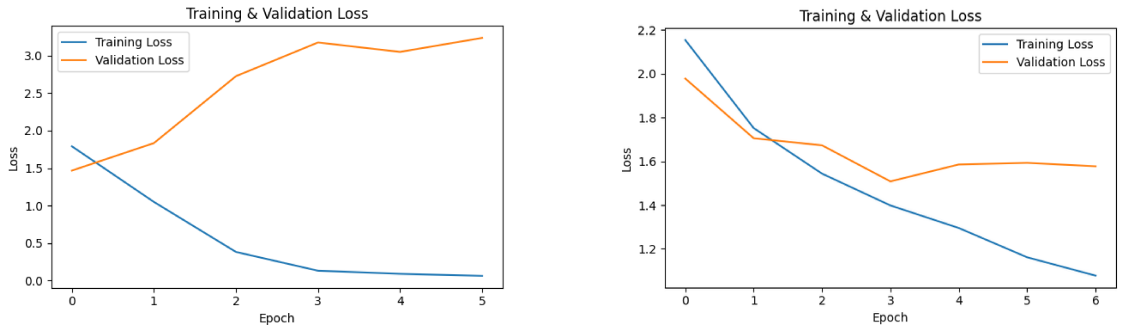

Figure 5.2: Effects of Dropout on CNN Training

5.3 XGB Classification Results

The final tuned XGB classifier displayed strong results, as shown in Figure 5.3, by combining several numerical and transform-based features such as wavelet decomposition (for multi-resolution) and MFCC for a compact audio representation. This combination of multiple approaches proved successful. Using multiple transforms and methods helped strengthen the feature space and provide a solid basis for classification.

5.4 Noise removal and Segmentation

Preprocessing the audio signal with noise removal and VAD segmentation was critical for achieving high accuracy. Figure 5.4 shows a screenshot of the Audacity software [77] illustrating the impact of these steps in clarifying and isolating the most relevant parts of the bird call.

Noise reduction is first applied to remove background noise, clarify bird song and help differentiate song from background silence. Then, the VAD segmentation was used to isolate the sections containing bird calls, removing stretches of background noise before and after. Performing noise removal first was essential to allow the VAD function to work more accurately.

The top track displays the raw audio signal for sample XC19922.ogg before being processed in the Audio Preprocessor pipeline developed in this study. The bottom track shows the resultant file, two equal

Classification Report:

	precision	recall	f1-score	support
0	0.48	0.48	0.48	200
1	0.57	0.47	0.52	200
2	0.64	0.68	0.66	200
3	0.59	0.71	0.65	200
4	0.55	0.61	0.58	200
5	0.78	0.83	0.81	200
6	0.78	0.79	0.78	200
7	0.73	0.59	0.65	200
8	0.73	0.66	0.69	200
accuracy			0.65	1800
macro avg	0.65	0.65	0.65	1800
weighted avg	0.65	0.65	0.65	1800

```
The Train Score is 0.8336111111111111
The Test Score is 0.6477777777777778
Confusion Matrix:
[[ 97  27   6  23  14   7   4   6  16]
 [ 41  95   7  12  29   0   9   4   3]
 [ 13   5 135  10  10  15   3   7   2]
 [  7  12   3 142  14   2   9   4   7]
 [  6  10   8  24 123   8  11   7   3]
 [  4   0  10   8   3 166   1   8   0]
 [  7   5  10  10   3   3 157   4   1]
 [ 12   6  23   4  12   7   1 119  16]
 [ 15   8  10   6  14   4   6   5 132]]
```

Figure 5.3: XGB Final Trial Results

five-second segments. Comparing the plots shows a considerable amount of noise reduction, as told by the peaks of the waveform and the lower peaks in the processed tracks. The red boxes have been added to highlight the approximate five-second windows segmented for training.

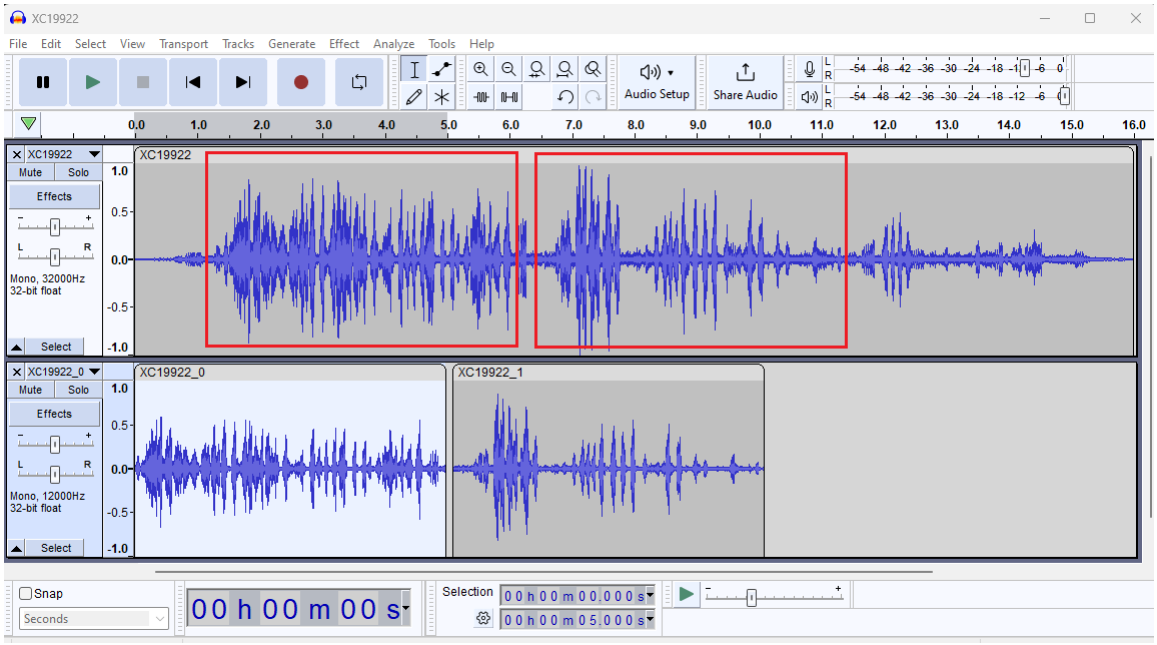

Figure 5.4: Audacity Example [77]

Reducing noise helped the models discriminate the dominant and most crucial bird call features. The segmentation helped ensure empty audio outside the calls did not deteriorate learning and improved efficiency as important audio sections were concentrated.

The next chapter, Discussion, interprets these results, acknowledges limitations, and suggests future work.

Chapter 6

Discussion

This chapter critically discusses the classification results from Chapter 5, relating them to the relevant literature in Chapter 3 and the methods outlined in Chapter 4. The analysis also connects the findings to the original research aims, objections, and questions stated in Chapter 1.

6.1 Research Objectives and Methodology

The primary objective of this research was validating and extending the CWT-CNN approach of Kiskin et al. [34] for mosquito detection to bird species classification, replacing the high-quality, controlled Mosquito recordings with the noisy, crowd-sourced Kaggle data. Their recommendations guided the experimental design, notably the CWT feature extraction and few-layer CNN architecture.

Another aim was to compare the CWT-CNN to a traditional classifier using manually-crafted features. Shift-invariant wavelet features recommended by Selin et al. [68] were combined with transforms popular in the BirdClef competition [3, 72, 4] to train a gradient boosting algorithm, a famous top performer in prior contests as noted by the Kaggle community [33].

Contrary to expectations, the gradient-boosting classifier outperformed the CWT-CNN on this dataset.

6.2 Comparative Analysis

6.2.1 Deep Learning vs Traditional Classifiers

As discussed in the literature review, deep learning has shown incredible success in bioacoustic classification [82, 81]. However, most studies manually inspect data, providing models with clean samples. They also restrict variability, only using one call type per species [68], giving them significant control over the quality of the raw training material and a clear advantage over this work.

Manually screening the 16.9k samples here (over 670k seconds or 180 hours) was infeasible, given this projects intentional aims to avoid biases from selective data curation intentionally.

The gradient-boosting classifier (XGB) achieved a 65% F1 score, outshining the CWT-CNN's 53% performance on unseen test data. While below expectations, the CWT-CNN exceeded the 50% project benchmark and improved with more data, demonstrating effectiveness for classification. Apparent overfitting aligned with the literature that deep learning requires ample training data [88], albeit these effects were reduced with the dropout countermeasures.

Overall, the XGB's superior F1 score and in-species classification results demonstrate the importance of traditional classifiers, mainly when working with noisy public datasets, challenging the popular notion that deep learning is always the optimal choice.

6.2.2 Relation to Previous Work

While the success of the XGB model aligned with competition expectations and prior history [33], the performance of the CWT-CNN did not reach the same heights as the deep learning studies of chapter 3. This sub-optimal performance may be attributed to the challenges posed by the dataset, being far noisier

and of lower quality due to its collection in the wild via a range of devices. This work confirms and contradicts the published results, demonstrating the difficulty of applying machine learning techniques to real-world audio signals and highlighting the potential.

Increasing to three convolutional and two dense layers boosted accuracy, aligning with Zhang et al.'s [87], whose highly successful single feature identification model included eight convolutional layers. However, after adding noise removal and voice activity detection (VAD) segmentation preprocessing, reducing layers to mitigate overfitting and improve training time, showed simpler networks can match complex ones post-processing.

Increasing wavelet scales boosted accuracy by generating more training coefficients and capturing higher signal resolution. However, redundancy emerges with extra near-zero coefficients, potentially increasing training time without benefit. Simple thresholding did not improve efficiency, though overfitting reduced, suggesting it may enhance sample quality by decreasing variability. While redundant coefficients can aid information recovery [17], this highlights the delicate balance in threshold tuning.

6.3 Performance Metrics and Results

6.3.1 Classifier and Feature Performance

The gradient boosting classifier achieved promising 81% F1 scores, indicating practical training on uncurated data is feasible. However, varying per-species performance from 28-74% F1 may reflect extreme sample noise and quality differences in the training data. The performance of CWT-CNN surpassed the 50% benchmark and demonstrated potential for enhancement with additional data, particularly if the training data was extended.

Experiments assessed additional preprocessing steps and methods to mitigate overfitting through an iterative model optimisation process within the strict time and resource project constraints.

The iterative CNN development provided insights into practical techniques and areas needing refinement. Results supported wavelets like Morlet and Gaussian, though the performance difference was insufficient to suggest a preference for either. Performance varied with architecture and parameters, presenting optimisation opportunities.

6.3.2 Overfitting and Regularization

Overfitting presented a particular challenge in this research, given the objective of limiting data to the 2023 BirdClef data only. Overfitting means that the model performs well on the training data but fails to generalise to unseen data, undermining the purpose of the model. In this work, the problem is even more severe given the variability and ambient noise in the wildlife recordings, aiding the model to learn noise patterns rather than bird call patterns.

Limited data inhibited overfitting, the severe impact of which was limited via the application of dropout, with scores increasing at 0.35 dropout probability. However, diminishing returns emerged at 0.40, with more extended training and negligible improvement. This suggests the regularisation techniques have restricted capacity with small datasets, needing tuning for optimal impact. Future work could explore alternative regularisation techniques, such as L1 or L2 regularisation, or even ensemble methods to improve the model's robustness.

In summary, the optimised CNN beat the baseline 50% F1 metric, showing valuable features can be learned from wavelets without complex feature engineering.

6.4 Data Quality

6.4.1 Impact of Data Quality

Throughout the CNN experiments, training accuracy remained higher than test accuracy, implying poor generalisation due to the small dataset. The effects of an imbalanced dataset were eliminated by setting the sample numbers equal from the outset of every iteration. However, this approach did not consider the quality of the recordings for each species or provide enough training data for the models. While data augmentation could potentially improve the situation, it goes against the project's goals. Instead, refining pre-processing techniques and scaling resources may still be a better way to address this issue.

Further analysis using VAD-based SNR calculations could confirm species sample quality variations. Given effective preprocessing, manual screening may be unnecessary, highlighting the need for objective

crowd-sourced audio ratings to reject poor samples. With a robust pipeline, results suggest labour-intensive curation could potentially be avoided.

The quality of the crowd-sourced data directly affected the classifier's performance. Bird call audio noise, and sparsity posed significant challenges but provided a realistic testbed for model accuracy. The impact of high-quality raw data cannot be overstated. It presents not only a challenge to projects but also a potential opportunity for future research aimed at enhancing quality with minimal resource.

6.4.2 Preprocessing Techniques

Manual inspection may offer the advantage of human expertise, but it will come at considerable cost and subjectivity, especially when dealing with large datasets. This subjectivity introduces potential bias, potentially skewing results and undermining the project's aims to avoid such measures.

The noise reduction and VAD were shown to be effective via manual inspection. The noise reduce algorithm and its associating parameters, such as the time constant and proportional decrease, proved effective in manual checks - however, the impact on many samples remains unknown. Relying on 5-second segmentation risks misrepresenting bird calls. The CWT's time-frequency resolution offsets this inaccuracy, but more sophisticated statistical segmentation methods may improve resolution.

The overall pipeline impact could not be quantified without an objective audio quality metric like SNR. In combination with a tested noise model, SNR could streamline the preprocessing pipeline, allowing for automated, unbiased selection or rejection of audio based on quality.

Overall, despite its variable and uncontrolled nature, results indicate that crowd-sourced data can be a valuable resource for species identification. They also suggest that traditional classifiers should not be overlooked, especially given their proven competitive successes under constraint. With training times typically of over four hours per experiment in the later stages of development, compared with twelve hours for the CNN models, these traditional classifiers trained on CPU provide positive encouragement that more subtle expert features may still prove successful over their deep learning counterparts.

6.5 Limitations

The results in Chapter 5 demonstrate the study successfully addressed the primary aims and research questions, showing that crowd-sourced data can enable meaningful scientific analysis. However, several limitations could have impacted the full potential of the methods used.

Computational constraints due to limited time and funding restricted the options to cost-effective solutions rather than an entire market exploration. Consequently, the study focused on nine species using only a subset of the available data, restricting the scope and general findings.

The quality of the raw audio, marked by high background noise and variability in recording methods, posed challenges for the classifiers. Furthermore, time constraints limited the fine-tuning of preprocessing and transform parameters. The absence of an accurate noise model and objective SNR-type metric further complicated these issues. Training accuracy consistently exceeded testing, indicating dataset size modelling constraints. Balancing classes addressed the imbalance but not the quality or quantity of the raw audio. Though honest in its pursuit, the research goal of unaugmented training potentially restricted the machine learning models from performing better.

The extended CNN training times and computational demands, significantly when increasing sample sizes and wavelet scales, caused exponential CWT computation growth compared to the mosquito study. Alternative resources, like cloud computing, may have better suited the feature pipeline.

These limitations by no means diminish the value of the research but rather serve as prompts for further investigation and act as essential caveats for the results.

6.6 Future Work

This research has provided valuable insights into applying CWT-CNN models to classify bird songs. Whilst results are promising, several areas of potential improvement and further analysis have been identified. This section outlines these areas and provides recommendations for future research.

6.6.1 Model Enhancements

- CWT Images: Instead of relying on coefficients, future studies could explore CWT images, possibly combining with pre-trained models like ResNet and ImageNet, which have demonstrated superior performance [3]
- Ensemble CNN Models: This study trained a single-channel CNN model. Constructing ensemble CNN models using images from scalograms, spectrograms, and other transforms could mitigate the weakness of an individual method and enhance classification accuracy.
- CWT Thresholding: Exploring more sophisticated thresholding methods, such as the SureShrink and BayesShrink (assuming prior distributions for the signal and noise) and Adaptive Thresholding (where different threshold values are used for different subbands of the wavelet transform) [89].
- Exploration of CNN Architectures: The study employed a relatively simple CNN architecture. Diversifying CNN architectures with more layers and elements like transformers or attention mechanisms could be explored. Conducting a systematic GridSearch for optimal parameters could also improve the model.

6.6.2 Preprocessing Enhancements

The study used basic noise reduction and VAD techniques, which, whilst effective to some degree, leave room for more sophisticated approaches that could substantially improve performance and maintain the project ethos of never manually selecting data.

- Bird Frequency Analysis: A detailed analysis of frequency ranges and statistical tests can provide a more precise understanding of bird song characteristics. This information can then inform decisions made in the preprocessing pipeline, such as defining sampling rates, bandwidth, window sizes, and segmentation to assist in isolating relevant frequencies.
- Wavelet Choice Analysis: The study employed a general-purpose wavelet for the CWT using suggested literature findings. Given the optimal wavelet choice is far from settled and warrants further investigation, future work could experiment with different wavelet families and scales. By using an inverse transform, a reconstructed audio signal could be compared with the original, with the best approximation measured to select the best wavelet.
- Adaptive Audio Segmentation: Creating structured methods for segmenting recordings could simplify the preprocessing phase. This includes refining VAD technologies or machine learning algorithms to differentiate bird songs from silence, which could automate tasks like the manual annotation system used by Cornell. More advanced methods are available, such as Hidden Markov Model-based segmentation [26].
- Audio Rating System: Creating an unbiased and quantifiable method to evaluate audio quality would be extremely helpful for this research and the wider bioacoustic community, especially for curators of the Xeno-canto collection. This system could rely on various acoustic characteristics and act as a dependable way to screen out low-quality audio samples before they are used for training.

In summary, this research has explored bird song classification using CWT-CNN models on raw audio data unlike any other. Whilst the XGB classifier outperformed the CNN model in terms of F1 score, the deep learning method still demonstrated potential for improvement, particularly given more extensive datasets and higher computational resources. These findings challenge the notion that deep learning is always superior in classification, especially when dealing with noisy, real-world data.

The study re-establishes a clear need for data quality and preprocessing techniques, revealing that even public datasets can provide valuable scientific insights when managed appropriately. Despite limitations, the research objectives were primarily met, providing a robust platform for future improvements.

The research serves as a crucial reminder that traditional methods still have a role to play alongside the advanced machine learning techniques and that signal processing steps warrant far more attention, with the potential to improve classifier performance before advanced tuning of model parameters even begins.

The final chapter recaps the research goals and critical outcomes, placing them in the existing literature to highlight their significance and potential impact.

Chapter 7

Conclusion

This concluding chapter revisits the original research objectives and questions that motivated the study. It provides a condensed summary of significant findings and accomplishments. The research is positioned within the context of existing literature to emphasise its importance and potential impact.

This thesis explored CWT features for CNN classification models, the primary aim was to validate published accuracy results for identifying bird species in crowd-sourced audio data. The project was determined to avoid manual selection and curation of data samples, with the objective to implement an effective modelling process that could work with raw, unfiltered data. Additional goals included comparing the performance of CNN models with traditional classifiers. This work also aimed to investigate various signal processing techniques, design a TensorFlow pipeline, and train a CNN model within a specific timeframe. Further objectives were to minimise overfitting without relying on data augmentation and to develop a traditional classifier using established techniques for comparison.

The project successfully navigated computational and memory constraints through systematic experiments, using a bottom-up project management approach guided by published literature. The results demonstrate effective bird classification with limited, raw, randomly sampled data and constrained resources. The traditional classifier achieved the highest F1 score of 65% on test data, outperforming the CNN in per-species testing and aligned with published literature, with some species achieving as high as 81% F1 score. However, the CNN achieved a final F1 score of 57% in under 12 hours, meeting its F1 target score and timeframe requirements. Furthermore, at least three species were classified with an F1 score exceeding 70% using both classifiers, aligning with and exceeding some published results and BirdClef competition entrants.

The CWT-CNN approach proved effective using controlled, short-duration samples. This research complements the existing work by extending the CWT to longer, lower-quality recordings, thereby demonstrating the potential and versatility of the wavelet transform. It also showed that handcrafted features from conventional transforms, paired with a traditional classifier, outperformed deep learning models on this dataset, contrary to the literature findings discussed in chapter 3.

The noise reduction and VAD segmentation methods enhanced audio quality and decreased non-tweet time, improving classification performance based on trial results and visual inspection. However, optimal audio preprocessing parameters could not be explored due to time constraints. The project designed an end-to-end infrastructure aligned with the core objective. All components displayed proven functionality with room for advancement. Dropout optimised the CNN's generalised performance, confirming the technique's effectiveness.

While meeting primary aims, the study was limited to nine species due to computational constraints. This necessary compromise allowed for a more in-depth analysis of the CNN architecture and reasonable training and test times. Acknowledging the challenges encountered, particularly in building the TensorFlow pipeline, is essential. Memory allocation issues arose due to oversized batches, leading to crashes that required troubleshooting and adjustments. Additionally, incorrect tensor shapes and coefficient formats from the CWT feature extraction necessitated further refinements to the preprocessing stage. These setbacks consumed valuable time and incurred additional computational costs. Training times stretched to 11 hours per model, further complicated by disconnection and session issues when using COLAB, straining both time and financial resources and patience.

This research provides a solid basis for future studies that use crowd-sourced data for bioacoustic

classification, offering a comprehensive, end-to-end classification model-building process that can be easily adapted and scaled. With studies relying on manual inspection and so much publicly available data often mislabelled, this study has shown that valuable identification information can be extracted, demonstrating that even noisy, unstructured, crowd-sourced datasets can be a goldmine for scientific discovery. The research has also identified numerous areas for exciting exploration.

Expanding the number of species would improve the model's utility. Refining the preprocessing pipeline by fine-tuning the parameters using statistical analysis of the audio dataset could provide significant classification gains. Scaling up the computational resources could enable more complex models, hyperparameter tuning and efficient analysis.

Overall, this research met its extensive list of ambitious aims and opened many more questions than it answered, offering numerous future enquiries. As a pursuit of academic study, this research has underscored the joy of the 'scientific method' with many an emotional upturn and downturn experienced during its completion.

Despite the solitary nature of the work, which proved troubling at times, the project stands as a testament to the power of scientific enquiry, best enriched through collaboration with friends and colleagues.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Lukas P. A. Arts and E. L. van den Broek. The fast continuous wavelet transformation (fcwt) for real-time, high-quality, noise-resistant time–frequency analysis. *Nature*, 2022.

[3] Awsaf. Birdclef23: Effnet, fsr and cutmixup. https://www.kaggle.com/code/awsaf49/birdclef23-effnet-fsr-cutmixup-train, 2023.

[4] Awsaf. Birdclef23: Pretraining is all you need. https://www.kaggle.com/code/awsaf49/birdclef23-pretraining-is-all-you-need-train, 2023.

[5] Myron C. Baker and David M. Logue. Population differentiation in a complex bird sound: A comparison of three bioacoustical analysis procedures. *Ethology*, 109:223–242, 2003.

[6] Eira Bermúdez-Cuamatzin, Alejandro A Ríos-Chelén, Diego Gil, and Constantino Macías Garcia. Experimental evidence for real-time song frequency shift in response to urban noise in a passerine bird. *Biology Letters*, 7(1):36–38, 2011.

[7] John Bevis. A complete history of collecting and imitating birdsong, 2023.

[8] BirdWatching. The best birdsong apps. *BirdWatching Daily*, 2021.

[9] Jason Brownlee. Dropout for regularizing deep neural networks, 2019.

[10] Steven L. Brunton and J. Nathan Kutz. *Data Driven Science & Engineering: Machine Learning, Dynamical Systems, and Control*. Department of Mechanical Engineering, University of Washington; Department of Applied Mathematics, University of Washington, 2018.

[11] Joeri L. M. Bruyninckx. *Sound science: recording and listening in the biology of bird song, 1880-1980*. PhD thesis, Maastricht University, 2013.

[12] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang. Sensor network for the monitoring of ecosystem: Bird species recognition. *IEEE*, 2007.

[13] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang. Sensor network for the monitoring of ecosystem: Bird species recognition. *IEEE*, 2007.

[14] E Chesmore and E Ohya. Automated identification of field-recorded songs of four british grasshoppers using bioacoustic signal recognition. *Bull Entomol Res*, 94(04):319–330, 2004.

[15] BirdWatching Daily. The best birdsong apps. *BirdWatching*, 2021.

[16] I Daubechies. *Ten Lectures on Wavelets*, volume 61. Society for Industrial and Applied Mathematics, 1994.

[17] David L. Donoho. De-noising by soft-thresholding. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 41(3):613, May 1995.

[18] J.L. Dowling, D.A. Luther, and P.P. Marra. Comparative effects of urban development and anthropogenic noise on bird songs. *Behavioral Ecology*, 23(1):201–209, 2012.

[19] Dpakkrish. Birdclef eda and pre-processing. https://www.kaggle.com/code/dpakkrish/birdclef-eda-and-pre-processing, 2023.

[20] eBird Help Center. How to rate media in the macaulay library/ebird, 2023. Modified on: Fri, 21 Apr, 2023 at 6:19 PM.

[21] Ulf Elman. Signal to noise ratio as a supportive tool in rating recording sound quality a-e, 2022.

[22] Ulf Elman. Weighted snr as a more robust and precise tool in rating perceived recording sound quality a-e, 2022.

[23] British Trust for Ornithology. Bto's state-of-the-art equipment and software, 2023.

[24] V. Georgieva, P. Petrov, and D. Zlatareva. Medical image processing based on multidimensional wavelet transforms. *Journal of Physics: Conference Series*, 2021.

[25] T Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. https://pypi.org/project/pyAudioAnalysis/, 2014.

[26] Theodoros Giannakopoulos. 5. segmentation, 2023.

[27] Herveŏ301 Goeŏ308au, Herveŏ301 Glotin, Willem-Pier Vellinga, Robert Planqueŏ3013, and Alexis Joly. Lifeclef bird identification task 2016: The arrival of deep learning. *CEUR Workshop Proceedings*, 1609:1–7, 2016.

[28] Google. Google colaboratory, 2022. Online platform for machine learning and data science.

[29] A Haar. Zur theorie der orthogonalen funktionensysteme. *Annals of Mathematics*, 69(3):331–371, 1909.

[30] Kaggle. Birdclef 2023 competition leaderboard, 2023.

[31] Kaggle and Cornell Lab of Ornithology. Birdclef 2023: Identify bird calls in soundscapes, 2023.

[32] A. Karpathy, G. Toderici, Sanketh Shetty, Thomas Leung, R. Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[33] YURY KASHNITSKY. Topic 10. gradient boosting, 2020. Kaggle Notebook.

[34] I. Kiskin, D. Zilli, Y. Li, M. Sinka, K. J. Willis, and S. J. Roberts. Bioacoustic detection with wavelet-conditioned convolutional neural networks. *Neural Computing and Applications*, 32:915–927, 2018.

[35] L. Koch and J. Huxley. *Songs of Wild Birds*. Witherby, 1938.

[36] Ludwig Koch. The first wildlife recording, 1889.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[38] The British Library. The british library sound archive, 2023.

[39] LifeClef. Lifeclef 2014 bird task, 2014.

[40] Zhang & Zhang Liu, Yao. Wavelet scattering transform for ecg beat classification. *Computational and mathematical methods in medicine*, 2020, 2020.

[41] S Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[42] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[43] J. Manzanares-Martinez, C. I. Ham-Rodriguez, and B. Manzanares-Martínez. Recovery of transit times and frequencies of multiple pulses via the short-time fourier transform. *Revista Mexicana de Física*, 2018.

[44] Grant McCreary. Review: Song sleuth app, October 2017. The Birder's Library.

[45] Y Meyer. Ondelettes et fonctions spline. *Hermann*, 1990.

[46] Hasanain J. Mohammed and Ali M. Al-Rahim. Processing ambient noise using wavelet analysis tools: the iraqi seismological broadband network data at iraqi meteorological organization and seismology (imos). *Kuwait Journal of Science*, 2023.

[47] J Morlet and A Grossmann. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.

[48] Jayme Moye. Testing out song sleuth, a new app that identifies birds by their calls. *Audubon*, 2023.

[49] Erwin Nemeth and Henrik 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*, 280:20122798, 2013.

[50] S. Newman, Aleksei A. Chmura, Kathy Converse, A. M. Kilpatrick, N. Patel, E. Lammers, and P. Daszak. Aquatic bird disease and mortality as an indicator of changing ecosystem health. *Marine Ecology Progress Series*, 352:299–309, 2007.

[51] Noisereduce. Noisereduce: An active noise reduction tool in python. https://pypi.org/project/noisereduce/, 2023.

[52] Cornell Lab of Ornithology. Birdclef 2023: Identify bird calls in soundscapes, 2023. Kaggle competition organised by Cornell Lab of Ornithology.

[53] The Cornell Lab of Ornithology. Identify bird songs and calls with sound id. *Merlin Bird ID - bird identification help and guide for thousands of birds*, 2023.

[54] S Parsons and G Jones. Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *J Exp Biol*, 203(17):2641–2656, 2000.

[55] Gianni Pavan, Gregory Budney, Holger Klinck, Hervé Glotin, Dena J. Clink, and Jeanette A. Thomas. *History of Sound Recording and Analysis Equipment*. SpringerLink, 2022.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python, 2011.

[57] App Store Preview. Merlin bird id by cornell lab, 2023.

[58] App Store Preview. Song sleuth bird song analyzer, 2023.

[59] Nirosha Priyadarshani, S. Marsland, and Isabel Castro. Automated birdsong recognition in complex acoustic environments: a review. *Journal of Avian Biology*, 49(5):jav–01447, 2018.

[60] Ricardo Ramos-Aguilar, J. A. Olvera-López, Iván Olmos, S. Sánchez-Urrieta, and Manuel Martín Ortiz. Parameter experimentation for epileptic seizure detection in eeg signals using short-time fourier transform. *Research in Computing Science*, 2019.

[61] R Rao. Xeno-canto: Bird recordings (extended) a-m. https://www.kaggle.com/datasets/rohanrao/xeno-canto-bird-recordings-extended-a-m, 2021.

[62] G.B. Reynard. The rediscovery of the puerto rican whip-poor-will. *Living Bird*, 1:51–60, 1962.

[63] M. Rhif, A. Ben Abbes, I. Farah, B. Martínez, and Yanfang Sang. Wavelet transform application for/in non-stationary time-series analysis. *Applied Sciences*, 9(7):1345, 2019.

[64] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38, 1991.

[65] Tim Sainburg. noisereduce 2.0.1, 2019.

[66] A. Salehi, Shakir Khan, Gaurav Gupta, B. Alabduallah, Abrar Almjally, Hadeel Alsolai, Tamanna Siddiqui, and Adel Mellit. A study of cnn and transfer learning in medical imaging: Advantages, challenges, future scope. *Sustainability*, 15(7):5930, 2023.

[67] scikit-learn developers. sklearn.ensemble.gradientboostingclassifier, 2023.

[68] A Selin, J Turunen, and J Tanttu. Wavelets in recognition of bird sounds. *EURASIP Journal on Advances in Signal Processing*, pages 1–9, 2007.

[69] Shazam. Shazam - music discovery, charts & song lyrics, 2023.

[70] B. Walther Shih-hsiung Liang, C. Jen, Chao-Chieh Chen, Yi-Chih Chen, and B. Shieh. Acoustic preadaptation to transmit vocal individuality of savanna nightjars in noisy urban environments. *Scientific Reports*, 10:18159, 2020.

[71] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

[72] Hayden Smith. Birdclef 2023: Deep audio classification. https://www.kaggle.com/code/haydenismith/deep-audio-classification-birdclef-2023, 2023.

[73] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[74] Natural State. Natural state, 2023.

[75] Wayne Storr. Band pass filter - passive rc filter tutorial, 2023.

[76] Jonathan Swift. *Gulliver's Travels into Several Remote Nations of the World*. Project Gutenberg, 1726.

[77] Audacity Team. Noise reduction, 2023.

[78] Librosa Development Team. librosa.feature.rms, 2023.

[79] Marketing Team. Bottom-up vs. top-down project management. *SaaS BPM*, 2022.

[80] PyWavelets Development Team. Continuous wavelet transform (cwt), 2023.

[81] B.P. Toth and B. Czeba. Convolutional neural networks for large-scale bird song classification in noisy environment. In *Proceedings of the Conference and Labs of the Evaluation Forum*, pages 1–9, Évora, Portugal, September 2016.

[82] T. Turker, A. Erhan, and D. Sengul. Multileveled ternary pattern and iterative relieff-based bird sound classification. *Appl. Acoust.*, 176:107866, 2021.

[83] A. Tyagi and Ritika Mehra. Intellectual heartbeats classification model for diagnosis of heart disease from ecg signal using a hybrid convolutional neural network with goa. *SN Applied Sciences*, 3(2):1–18, 2021.

[84] S Winkler. Perceptual distortion metric for digital color video. In *Proc. SPIE*, volume 5666, pages 175–184, 2004.

[85] J. Xie, M. Towsey, J. Zhang, and P. Roe. Frog call classification: a survey. *Springer*, 2016.

[86] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9:611–629, 2018.

[87] F. Zhang, L. Zhang, H. Chen, and J Xie. Bird species identification using spectrogram based on multi-channel fusion of dcnns. *Entropy*, 23(11):1507, 2021.

[88] Tianyi Zhang, Cuiyun Gao, Lei Ma, Michael Lyu, and Miryung Kim. An empirical study of common challenges in developing deep learning applications. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pages 104–115. IEEE, 2019.

[89] Xiao-Ping Zhang and M. Desai. Segmentation of bright targets using wavelets and adaptive thresholding. *IEEE Transactions on Image Processing*, 10(7), 2001.

[90] Yudong Zhang, Jie Wang, Shuihua Wang, and Lei Guo. Forecasting stock indices with backpropagation neural network based on wavelet decomposition and phase space reconstruction. *PloS one*, 15(1):e0227222, 2020.

[91] Morgan A Ziegenhorn, K. Frasier, J. Hildebrand, E. Oleson, R. W. Baird, S. Wiggins, and S. Baumann-Pickering. Discriminating and classifying odontocete echolocation clicks in the hawaiian islands using machine learning methods. *PLOS ONE*, 2022.

Appendix A

Nine Species for Classification

Barn Swallow

Common Buzzard

Common House-Martin

Common Sandpiper

Eurasian Hoopoe

European Bee-eater

Thrush Nightingale

Western Yellow Wagtail

Willow Warbler

Appendix B

AudioPreprocessor Audit Log

Table B.1: Audio Preprocessor Version Control

Version	Date	Comments
v1	15/07/2023	Load audio and process in batches
v2	17/07/2023	Refactored, detect_silence added, high_pass filter
v3	19/07/2023	Remove silence (leading and trailing only)
v4	27/07/2023	Added trim and removed batch_size (wasn't being used anymore)
v5	31/07/2023	Parameters for noise_reduction (e.g. prop_decrease)
v6	01/08/2023	Reduce noise (spectral gating) added, order changed
v7	02/08/2023	Remove_silence upgrade, segment, trim the silence and rejoin
v7a	03/08/2023	Chop 5s segments, produce labels and copy of the dataframe rows
v8	13/08/2023	Tweaked noise_reduce using non-stationary and small window
v8	13/08/2023	Have added silence removal using the pyAudioAnalysis method

Table B.2: Final Audio Preprocessing Parameters

Name	**Value**	**Comment**
df_sample	df	Metadata file for samples
input_dir	INPUT_DIR	Directory for input files
output_dir	OUTPUT_DIR	Directory for output files
down_sample	12000	Downsampling rate in Hz
lowcut	250	High-pass filter cutoff
highcut	5950	Low-pass filter cutoff
gain	1.0	Amplitude adjustment
top_db	20.0	Silence threshold
apply_filter	True	Apply bandpass filter
apply_noise_reduction	True	Apply spectral gating
stationary	False	Stationarity flag
prop_decrease	1.0	Proportional decrease
apply_remove_silence	True	Remove leading and trailing silence
apply_detect_silences_waveform	True	Detect silence within waveform
silence_threshold	0.0001	Silence detection threshold
silence_duration	0.5	Silence duration in seconds
silence_margin	0.001	Silence margin
apply_normalise	False	Apply audio normalization
norm_type	'standard'	Normalization type
trim_audio	False	Flag for audio trimming
trim_length	5	Length for audio segments

Appendix C

CWT Thresholding Visualisation

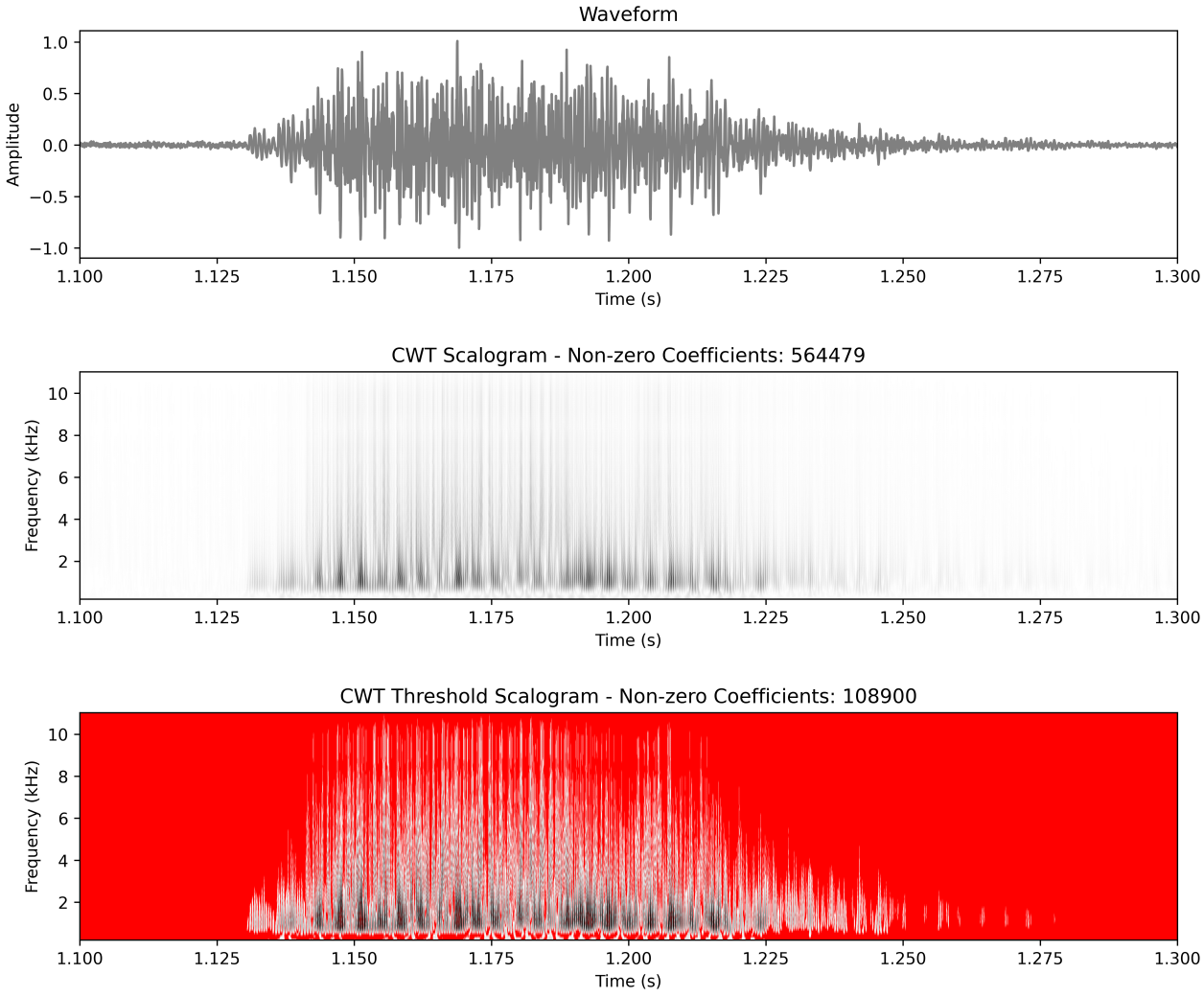

Figure C.1: CWT Thresholding Example

Appendix D

CNN Variables and Parameters

Table D.1: CNN Model Parameters

Name	Value	Comment
General Parameters		
Model Name	CNN	Convolutional Neural Network
CNN Layer Parameters		
First Conv2D Layer	16, (3, 3)	16 filters, 3x3 kernel
First MaxPooling	(2, 2)	Pooling size 2x2
Second Conv2D Layer	32, (3, 3)	32 filters, 3x3 kernel
Second MaxPooling	(2, 2)	Pooling size 2x2
Flatten	-	Flattening layer
First Dense Layer	128	128 neurons, ReLU activation
Output Layer	9	9 neurons, softmax activation
Other Parameters		
Dropout Rate	0.45	Rate for dropout layers
Regularization	L1=0.001, L2=0.001	L1 and L2 regularization

Table D.2: Variables and Parameters for Bird Sound Classification

Name	Value	Comment
Model Name	CNN	Convolutional Neural Network
`wavelet_type`	'morl'	Type of wavelet used
`fs`	12000	Sampling rate in Hz
`frequencies`	np.linspace(6000, 200, 64)	Frequency range
`scales`	Derived	Converted from frequencies
`batch_size`	10	Number of samples per batch
`num_classes`	9	Number of output classes
`epochs`	20	Number of training epochs
`prob_drop`	0.45	Dropout rate
`initial_learning_rate`	0.001	Initial learning rate
`decay_steps`	100000	Decay steps for learning rate
`decay_rate`	0.96	Decay rate for learning rate
`l1`	0.001	L1 regularization term
`l2`	0.001	L2 regularization term
`early_stop_patience`	3	Patience for early stopping
`optimizer`	Adam	Optimization algorithm
`loss`	'categorical_crossentropy'	Loss function
`metrics`	Various	Accuracy, Precision, Recall, AUC, F1Score
`checkpoint_path`	'best_model.h5'	Path to save the best model

Appendix E

XGBoost Development Cycle, Variables and Parameters

Table E.1: XGB Iterative Development Cycle

TL	AP	n	EST	SUB	DWT	STFT	WPD	MFCC	CV	TRAIN	TEST	TEST_F1
1	v4	70	10000	-	db10	2048	-	-	-	1.00	0.43	-
1b	v5	230	10000	-	db10	2048	-	-	-	1.00	0.46	-
2c	v5	400	10000	-	db10	2048	-	-	-	0.83	0.48	-
3a	v6	300	5000	-	db10	512	4	-	5	0.87	0.50	0.50
4	v8	800	5000	-	db10	512	4	-	5	0.75	0.37	0.37
5	v8	540	5000	0.90	db10	512	4	13	5	0.85	0.56	0.56
7	v8	1000	5000	0.90	db10	512	4	13	5	0.83	0.65	0.65

Table E.2: Variables and Parameters for Bird Sound Classification

Name	**Value**	**Comment**
Model Name	GradientBoostingClassifier	Traditional Classifier for comparison
`test_size`	0.2	20% of data used for testing
`random_state (split)`	42	Seed for data split
`n_estimators`	5000	Number of boosting stages
`learning_rate`	0.1	Rate of learning each stage
`max_depth`	3	Maximum depth of trees
`validation_fraction`	0.2	Fraction for early stopping
`n_iter_no_change`	10	Iterations with no improvement
`tol`	0.01	Tolerance for early stopping
`random_state (model)`	42	Seed for model training
`subsample`	0.9	Fraction of data for boosting
`cv (cross-validation)`	5	Number of folds in k-fold
`scoring`	'accuracy'	Metric used for evaluation