M.Sc. Thesis

Bird voice classification using Mask Region-based Convolutional Neural Network

Scholar

Payal

Student ID: 00646914

Supervisor

Dr. Chris J Hughes

Lecturer

School of Science, Engineering, and Environment
University of Salford
Manchester

September 2022

Bird voice classification using Mask Region-based Convolutional Neural Network

Scholar

Payal

Student ID: 00646914

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Data Science at the University of Salford Manchester

Thesis Supervisor

Dr. Chris J Hughes

Lecturer

Thesis Supervisor: _______________________________

Title

School of Science, Engineering, and Environment
University of Salford
Manchester

September 2022

Abstract

Our everyday lives depend heavily on our ability to hear. Numerous research has been conducted in recent years to translate this skill to computers. To help the computers classify various sound occurrences, we build and apply deep learning-based algorithms in this dissertation. In this thesis, we will evaluate and research the classification of bird sounds, which distinguishes between different bird species based on their sound events and event types. We employed deep learning techniques, such as region-based convolutional neural networks and deep convolutional neural networks, to categorize and identify different bird species based on their sounds. With this technique, we consider the sound's Spectrograms as images and turn the challenge of recognizing temporal bounds for a bird's sound type into the problem of detecting things in pictures. The log-scaled MEL-Spectrograms of the sound files are first used by the Deep Learning-based model to Mask R-CNN, a technique for recognizing objects in pictures. After that, we send these cropped sections to a straightforward CNN classifier, which will categorize them according to the type of bird sound they belong to. The classifier model, which was independently trained, will examine the scaled MEL Spectrogram's extracted region. Our analysis shows that this method may be applied in a bird sound classification system with innovative outcomes and correctly identify the bird’s sound with bounds on that particular portion of the sound.

Key words:

Deep learning, Bird Songs, Mask-RCNN, Machine Learning, multiclass classification, MEL Spectrogram.

Undertaking

I certify that the research work titled — "Bird voice classification using Mask Region-based Convolutional Neural Network" is my work. The work has not been presented elsewhere for assessment. Due reference and acknowledgment have been made, where necessary, to the work of the others.

Signature of Student
Payal
MSc-Data Science: 00646914

Dedication

I dedicate this research work to my loving parents, who always pray for my success and support me at every stage of my life; my brothers and sister, who always demonstrate concern for my future. May God bless them with good health and long life.

Acknowledgment

I have all the praises to the Lord of the whole universe, who made me the super creature, blessed me with knowledge, and enabled me to accomplish this research work. I wish to express my profound gratitude and indebtedness to my advisor, ***Dr. Chris J Hughe.***, Lecturer School of Science, Engineering and Environment University of Salford Manchester, UK for providing inspirational guidance, support, encouragement, and valuable suggestion throughout my research work. I am deeply grateful to ***Mrs. Mo Saree***, course leader in the School of Science, Engineering and Environment University of Salford Manchester, UK, for providing guidance, and valuable technical support. I am thankful to ***Mrs. Geoff Carss,*** *and* **Lorenzo's Trojan** they offered me to work on Machine learning project and I am very pleased for their guidance, assistance, and suggestions throughout this project. I had learn atlot under their guidance and support and curently they are working with Wilder Sensing Technology as a lead partner. I am very thankful to my loving father **Pritam Singh Saharan** and **Smt. Darshana Devi**, for their guidance, support, and much care and love throughout my academic career. My sincere gratitude also extends to all my friends and class fellows who have patiently extended all sorts of help for accomplishing this study successfully

Table of Contents

List of Figures

List of Tables

1 Chapter 1: Introduction

1.1 General Overview

The ability to recognize sounds is one of the most significant human traits. This ingrained human memory is an ability that supports us in our daily life to recognize a particular sound and link it with particular action (Sharda and Singh 2012) (Tomasino et al. 2015)

For data processing applications, artificial intelligence has gained a lot of attention in the past few years. A large amount of data and high computational power is required for such models to find trends in the data. Because of innumerable parameters arranged in successive order, such frameworks are known as deep learning models. The recent boom in deep learning has significantly impacted signal processing a lot. High computational power and a large number of labeled datasets have made deep learning models leave conventional methods behind (Purwins et al. 2019).

One of the initial methods of deep learning includes sound signal processing specifically in voice recognition(Hinton et al. n.d.). Soon the environmental sound assessment followed the same direction by embracing the most recent developments (Cakir et al. 2015) (Hershey et al. 2017). This study involves a variety of such processes. Sound incident detection is one of its kind, in which the incident is determined by its sound (Kumar and Raj 2016). Sound occurrences in this scenario relate to clear sounds that are hearable from background noises. The sound data consists of two types of labels for two different incident recognitions. The first labels are considered strong compared second labels which are weak. The weak labels only determine the existence of the incident in the sound while in its contrast strong labels specify the time boundary of the incident. It is difficult to recognize sounds that are labeled strongly. Accurately annotated sounds can be highly-priced which is another primary reason it typically needs human aid to determine when an incident occurred or ended.

Categorization of environmental sounds that classify incidents depending on sound signals is one of the most common tasks while processing non-speech sound signals (Chu, Narayanan, and Kuo 2009) (Valero and Alias 2012)(GnaMELe et al. 2019). Including monitoring systems, navigating robots, and autonomous vehicles there are many applications relying on the classification of environmental sounds (Chandrakala and Jayalakshmi 2020)(Dobre et al. 2015). Recognizing the sounds of a chainsaw in the forest is yet another part of recent applications of environmental sound categorization (GnaMELe et al. 2019). Recognizing sounds like shouting and screaming in domestic as well as in public is yet another useful application of the classification of sounds. Accurately recognizing a number of these sound occurrences could be extremely difficult because they resemble noise in some ways.

Surrounding us we see many different birds flying, identifying the type and specie of bird in any certain area depending on its recorded sound is the initial way to determine its specie because frequently birds produce distinctive calls and songs that set them apart from other birds (Brandes n.d.). For example, the forest is known as home to many animals and among these several animals, birds are one of them. For communicating with one another in their flock, birds use their sounds. Birds produce different sounds for different purposes. Including calling the flock, throughout breeding season luring birds of the opposite sex, and alert of any danger whenever threatened (Zhang et al. 2019). These sounds must be studied in detail to identify the circumstance behind the sound.

Unfortunately, annotating dozens of species of birds manually is labor-intensive, challenging, and prone to error. Technological breakthroughs in object detection and sound signal analysis can enhance sound classifiers that can accurately label and categorize different sounds of birds. Additionally, several yearly contests for bird recognition, including BirdCLEF, offer insightful information on how to effectively prepare the data and create reliable sound identification algorithms (Kahl et al. 2017). Because of recognizing primary signals (animal sounds) apart from the background noises, such events focus more on difficult bioacoustics tasks which typically require the ability to distinguish them. for this challenge, 30,000 labeled recordings of 1500 different species were used but when this labeled dataset is reduced to 30 sounds per specie it becomes more challenging. This is where deep learning helps.

1.2 Aims and Objectives

This study uses deep learning algorithms to include bird songs and metadata from the Xeno-Canto database to examine the identification and categorization of bird species. Handling class imbalance issues in various species class classification issues would help to improve and address the classification results.

The objectives involved in this study are:

- To design models for the identification of different species by sounds, MEL spectrogram features were collected from the Xento-Canto database.
- To test the model on new data on deep learning techniques for the identification of different bird species from their sounds.

- To use deep learning to conduct experiments and various data analysis techniques.
- • To create a solid model that can compare the best predictions for voice categorization of different bird species.
- • To investigate how various elements, such as the environment, a species' distinctive sound, and a voice recorder, impact birds found on many continents of the globe and how they are more common under certain conditions.

1.3 Scope of Research Work

Recent developments in deep learning and data science, which have sped up research on metadata, have significantly increased the relevance of the analysis of bird species and their numerous subspecies. A big challenge is maintaining the linkages between many research disciplines to their new topic and domain. Understanding different bird species is essential for developing new strategies and methods for analyzing data for different bird species. In this thesis, we shall list the bird species that were gathered from various sources in the United Kingdom. To identify the many bird species that are involved in colonial migration and the movement of bird species in the various areas of the world, we perform data analysis and use a deep learning model called Mask-RCNN. By capturing their noises and observing how they navigate across diverse surroundings, we may better understand the many different bird species. Create a deep learning model that can identify and classify certain species based on the input information, such as speech recordings acquired from the Xento-Canto database. This analysis will allow us to use analytical reasoning to discuss the various bird species, which will be discussed in the method part. The model may be used to classify different bird species using information.

1.4 Tools for Data Analysis

For data analysis, we have employed a variety of programs and libraries. Python was picked as the language since it is open source, and many free-source modules enable us to do tasks like data cleaning, analysis, deep learning model training, audio data analysis, and graph plotting with complete information. Since Google Collaboratory offers free GPU training services, we also used it to create the deep learning model. The libraries that we utilized in this were for the following purposes:

- Pandas: it is used for the data reading and filtering of the data using different queries.
- Sklearn: to implement the machine learning model and to use the measuring parameter to evaluate the models
- Matplotlib: The plotting of the graphs has been done with the help of Matplotlib

- Seaborn: It is also used for plotting visual information
- Librosa: For audio mainupulating
- Pytorch: used for deep learning model and training purpose

2 Chapter 2: Literature Review

2.1 Related Work

Speech and non-speech are two types of research domains in sound analysis. Speech assessment involves listening to sounds that have spoken words from one or perhaps more people covering a variety of subjects. Denoising (Aggarwal et al. 2011) (Shrawankar and Thakare 2010) automatic speech identification (Juang and Rabiner 2004) (Yu and Deng 2015) speaker identification (Choi et al. 1996) emotion identification (Casale et al. 2008) (El Ayadi, KaMEL, and Karray 2011) and recognizing disease stages, such as Parkinson's, among others are some important research domains of speech analysis. Studies are being carried out in the area of speech processing to enable machines to comprehend spoken words. Additionally, it is possible to determine the state of some ailments, including Parkinson's and Alzheimer's, by examining the individuals' speech (Caliskan et al. 2017) (López-de-Ipiña et al. 2013). Non-speech sound researches include a signal of sounds apart from human sources, such as surrounding, traffic, music animals, and other living and non-living things. automatic sound transcription (Drossos et al. 2019) sound incident recognition and categorization (Atrey, …, and 2006 n.d.) (Heittola et al. 2013) (Radhakrishnan, …, and 2005 n.d.) acoustic scene categorization (Barchiesi et al. 2014)(Mesaros et al. n.d.) sound source differentiation (Vincent et al. n.d.) includes in the majority of the discussed area in non-speech processing. These topics are furthermore explained with greater depth because they are more in line with the purpose of our study.

2.2 Sound Representations

High dimensional signals are considered insignificant to be fed as the input because their raw form is more noise sensitive (Hoshen et al. n.d.; Jaitly, on, and 2011 n.d.; Palaz, Collobert, and Magimai-Doss 2013). This is increasingly common to employ smaller, less dimensional sound signals as inputs since they preserve just the most specific information. The most useful techniques for extracting features are discussed below.

2.2.1 Spectrogram

A useful approach to derive features using the amplitude value of a signal’s frequency domain is the short-time Fourier transform (STFT) (Sejdić et al. n.d.). Smaller portions of a signal are made by using a sliding window. Spectral leakage of a signal is avoided by Hamming or Hanning which is a window function that is multiplied by each signal inside an individual portion. A 2-D visualization known as a Spectrogram is created by retrieving the frequency of the sound portions using a sliding window. The Spectrogram's value represents the intensity of a particular frequency band at a given point in time from input sound. A sound wave with its derived Spectrogram is displayed in Figure 1.

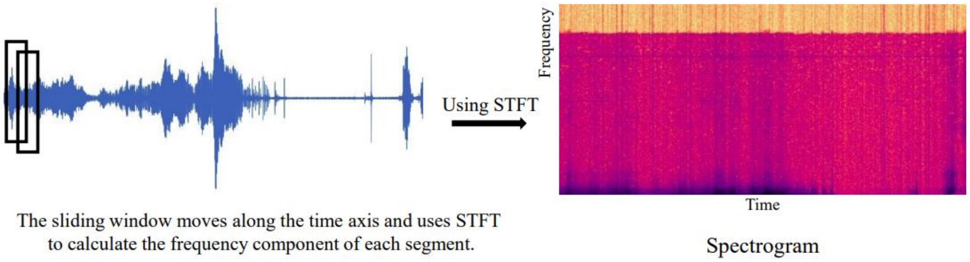
Using STFT
Frequency
Time
The sliding window moves along the time axis and uses STFT to calculate the frequency component of each segment.
Spectrogram

Figure 1. Sliding Window uses STFT to measure frequency component

2.2.2 Log-scaled MEL Spectrogram

Depending on the human auditory system, the MEL scale was developed(Stevens, …, and 1937 1937). Instead of the values with the higher frequency, it is more adept at recognizing variations in the values with lower frequency. To mimic the characteristic of the human sense of hearing, the MEL scale is developed by bands that become narrower and larger for lower and higher frequencies. MEL-Spectrograms are those which mix neighboring frequency bins using the MEL scale. Also for an average listener to interpret it as being twice as loud, the sound amplitude sound's amplitude needs to be nearly ten times higher. The MEL-Spectrogram's log scale is computed to take advantage of this property and bring sound representations similar to the human sense of hearing. A sample of a Spectrogram being transformed into a log-scaled MEL-Spectrogram is shown in Figure 2.

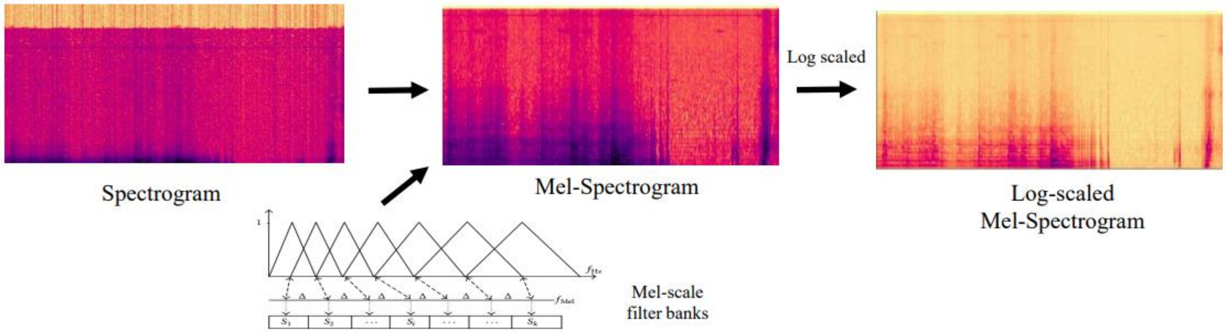

Figure 2. Sound wave analysis from Spectrogram to MEL-Spectrogram to the log of the MEL-Spectrogram

2.3 Traditional Machine Learning methods for Bird's Sound Classification

Random forests (RFs) (Phan et al. n.d.)(Zhang, Systems, and 2015 2015) hidden Markov models (HMMs) (Peng et al. n.d.) (Zhuang et al. 2010) support vector machines (SVMs), and Gaussian mixture models (GMMs) (Atrey et al. n.d.) are some of the traditional machine learning techniques utilized in this domain. A brief overview of the above-mentioned individual method is discussed below to define how they can be implemented to identify and categorize sound events. The probability density of each element of a composition is determined by generative models such as Gaussian Mixture Models (GMMs). The probability distribution of individual elements can be determined by a common technique known as expectation maximization (Moon 1996). Every category of sound serves as an element to identify the sound events and to recognize the characteristics relating to the distribution of individual sound classes in the model is trained. The temporal dependencies in the signal are not captured by GMM because every segment of the sound stream is handled independently. Contextual information is acquired to identify the sound events by implementing Hidden Markov Models (HMMs) (Heittola et al. 2013). An HMM classifies the present sound fragment while also taking into account prior sound fragments (Schuster-Böckler and Bateman 2007)

Differentiation techniques such as Support Vector Machine which splits the data sample through hyperplanes within high-dimensional space(Schölkopf 1998). SVM splits the sound stream into smaller fragments and classifies individual segments separately (Plinge et al. n.d.; Temko, Letters, and 2009 n.d.). SVM anyhow cannot deal with large datasets since it expands with the size of the dataset super-linearly.

Another machine learning technique that utilizes a group of decision trees relatively known as random forest (RF) is also suitable to perform categorization and regression tasks (Breiman 2001). When an RF model is trained using a portion of the training set separately for every decision tree. In the evaluation process, input labels are generated by every decision tree output, but the conclusion is determined by the majority voting method. A bunch of estimated features for an individual sound stream is utilized to train the RF model to recognize the occurrence of an incident in the stream in this way RF models categorize and identify sound events (Phan et al. n.d.; Zhang et al. 2015). Deep learning models have taken over classification detection tasks from machine learning models (Dang et al. n.d.; Xia et al. 2019). A brief overview of the neural network's layers is discussed later in this chapter.

2.4 Deep Learning

The way neurons in the human brain carry out different cognitive skills serves as inspiration for artificial neural networks (ANNs) (Schuster-Böckler and Bateman 2007). Each signal activates a different neuron in a human. Cognitive perception is generated when an input signal is translated by a neuron. An individual neuron in an Artificial Neural Network is an input signal's weighted value before passing the result by an activation function which may be linear or non-linear. Copying the human brain, dense layers are connected in between neurons allowing input signals to travel through and eventually map to resultant signals. The initial layer that obtains the input signal is known as the input layer likewise the final layer that generates the signal's output is called the output layer. The term "hidden layers" refers to every layer in between input and final layers. A multi-layer perceptron (MLP) method with two hidden layers is shown in Figure 3.

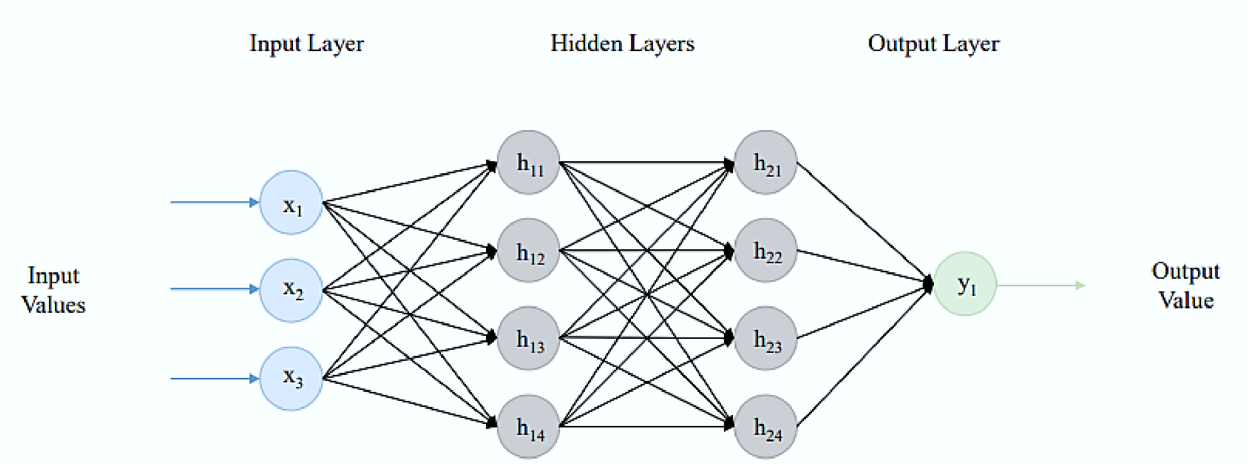

Figure 3. Multi-Layer Perceptron with 3 Input

The first linear model to distinguish between two types of data was developed by McCulloch and Pitts in 1943, which is where the ANNs got their start (McCulloch and Pitts 1943). Rosenblatt presented the perceptron, a pattern identification technique with two learning layers, in 1958 (Rosenblatt 1958). In (RuMELhart et al. n.d.) The weights of the connections in the method were changed using the backpropagation method, to reduce the discrepancy between the determined outcome and the desired result. In 2006 another major development came in, where Geoffrey Hinton demonstrated greedy layer-wise pre-training (Hinton et al. n.d.) of neural networks with multi-stacked layers. Deep learning or "deep neural networks" is a concept that has initiated to appear frequently related to neural networks that have more than one hidden layer. A deep network performs much more complex tasks by increasing the number of layers and units inside every layer(Anon n.d.-a). Large dataset requirements are one barrier to training deep models with vast numbers of parameters. Using a machine learning approach has a huge amount of labeled data that enables the model to train the mapping function using input samples for the output. Thanks to the capabilities of the internet generating large datasets are now easier. In 2009 (Dekkers et al. 2017) when ImageNet was released publicly, it already consisted of nearly 3.2 labeled data which now has reached about 14 million.

Presence of large datasets facilitated to train of deeper and more complex networks. The AlexNet is a network consisting of 3 fully connected layers and 5 convolutional layers, which won the ImageNet contest in 2012 by a significant margin over the runner-up (Krizhevsky et al. 2017). Right after that neural nets having more convolutional layers, as well as parameters, have significantly improved the efficiency in a variety of tasks. For instance, in 2016, a model consisting of 152 convolutional layers along with 60 million parameters was unveiled. The accuracy of classification on ImageNet has enhanced up to 78.3% due to RedNet-152. Such an accomplishment is even more remarkable when we consider that AlexNet obtained the most cutting-edge performance on ImageNet only four decades back.

2.4.1 Hyperparameters

The structure of the model is specified by hyperparameters, in addition to strategies required to train deep networks. The value of these hyperparameters (Ilievski et al. n.d.) can strongly affect the performance of the model. some of the significant parameters are listed below.

- Batch size is the number of training samples that can be taken as input while training the model. Depending on the loss signal, parameters on the model are identified. The value of that loss is defined by the difference between the expected outcome and the determined one. Resultantly a smaller batch size will cause unstable training of the model while batches may cancel one another which prevents the best training of the model.
- The learning rate determines the step size which is applied to modify the parameters of the model in individual iterations to reduce the estimated error. The training process can get delayed by learning rates that are too small, a model may also diverge or converge by a too large learning rate to parameter values that are not optimal.
- The number of hidden layers is strongly associated with the function’s complexity on which the model is trained, more complex relationships between input and output can be recognized by the higher number of layers. To train a deeper model, more data is also needed.

- The number of epochs determined the iteration number during the training stage. During each iteration, the model is introduced to an individual sample of the training set which modifies the parameters of the model following the determined loss value. Until stable performance and best parametric values are attained the iterations are repeated by the model. Overfitting of the model on the training set must be avoided. An overfit model has memorized pattern in the training set due to which it cannot categorize and cannot perform properly on test data. A smaller dataset and an increased number of iterations cause the model to overfit.

2.4.2 Fully Connected Neural Networks

A fully connected layer is linked with the individual activation value from the layer before it. Being "Structure agnostic" is the main benefit of these layers, in other words, they don't presuppose anything regarding input data structure (Anon n.d.-b)

Suppose the output to the *l-th* layer h^l having input h^{l-1} can be calculated in Equation 1:

$$h^l = \sigma(W^l h^{l-1} + b^l) \qquad (1)$$

σ in the above equation is the activation function whereas W^l and b^l are the weight and bias parameters. The non-linear, continuous, differentiable activation functions improve the ability to recognize the non-linear relationship between input and output values of the neural network. The four most typical activation functions are displayed in Figure 4.

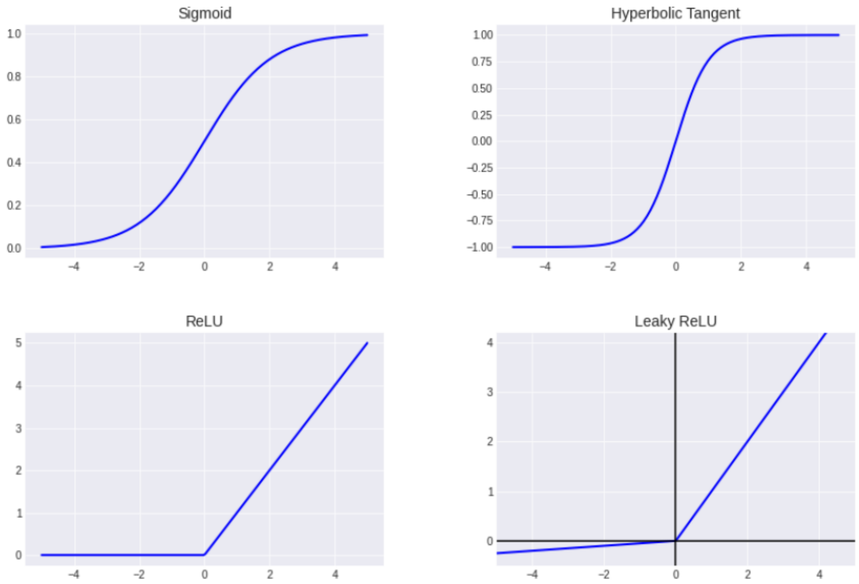

Figure 4. Some Popular Activation Functions

2.4.3 Convolutional Neural Networks

Initially convolutional neural networks were applied to image recognition tasks (Krizhevsky et al. 2017; Simonyan and Zisserman 2015). To convolve properly the input of these layers is stridden over by several kernels. Kernel's parameters are shared with several input elements while compared to fully connected layers it significantly reduces the number of parameters. When the individual kernel is convolved with the input a feature map showing the position and strength of a particular feature is generated. the size of this feature map is decreased and the identification of the features is made in the input invariant shift by applying pooling layers to the final layer(Scherer et al., 2010). Pooling layers such as Maxpool and Average pool are used most commonly to plot the maximum and an average number of values to the feature map respectively (Boureau et al. n.d.; Yu et al. 2016). Very lately, attention pooling, which assigns weights to various input components, has also been demonstrated to enhance the efficiency of the network (Vincent et al. n.d.).

Usually, while implementing convolutional neural networks, pooling layers are succeeded by convolutional layers. These models integrate the basic features that are detected in the initial layers to identify progressively sophisticated elements as the network is built up.

2.4.4 Recurrent Neural Networks (RNN)

To identify connections between the sequenced values, RNNs are frequently utilized in the analysis of sequential data, for example, text or sound (Graves et al. n.d.; Tom´ and Mikolov 2010). Recurrent layers incorporate the activations of earlier timespans as well as the present input. The RNN is a bi-directional network, in which the output of a given period is calculated using the present input as well as previous and future activations (Schuster and Paliwal 1997)

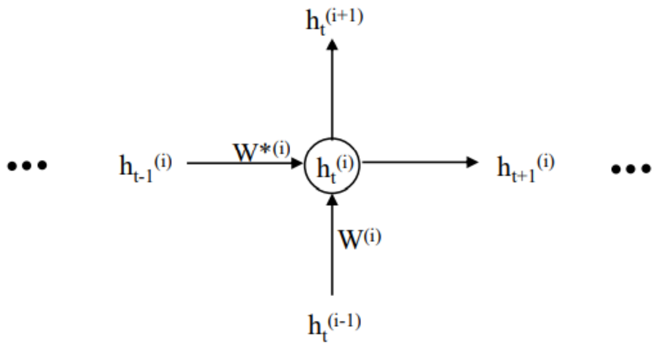

Figure 5. Recurrent Neural Layer

Suppose in implementing a usual RNN in which at time t, hti-1 is the input given to the ith hidden layer for which ht(i) can be calculated in Equation 2:

$$h_t^{(i)} = \sigma(W^{(i)} h_t^{(i-1)} + W^{*(i)} h_{t-1}^{(i)} + b) \qquad (2)$$

The weight matrix between i-1 and I is denoted by W(i) while W*(i) is the output layers' weight matrix I for the previous time. The recurrent neural layer in Figure 5 gets the activation of the preceding layer as well as the activation of the preceding time step to generate the output at time-step t.

3 Chapter 3: DATA MATERIAL AND SELECTION

3.1 Introduction

This chapter covers the detailed explanation of data material used in this thesis, Sounds Clips of the birds with metadata for the bird voice classification using Deep Learning techniques.

3.2 Xanto-Cento Database

A platform xeno-canto (Vellinga, Notes), and 2015 n.d.) is devoted to disseminating bird sounds from all around the world. We urge you to listen to, download, and explore the collection of bird sound recordings, whether you are a researcher, a birder, or just interested in a sound you heard out your kitchen window. However, xeno-canto (Vellinga et al. n.d.) is more than just a library of audio files. As a team effort, it is a project. And everyone can contribute to this project which will assist to solve the bird's songs puzzle recording. Below are some of the details of the dataset which will be used in this bird voice classification using the Deep learning methods.

3.3 Statistics

Here are some fascinating facts about the xeno-canto collection. The contributors are continuously adding new recordings from time to time which will be helpful for the researcher to solve many problems related to bird's sound analysis.

3.3.1 Collection Graphs

Several graphs illustrate the collection's development throughout time. For recordings posted during the first several years of xeno-existence, canto's upload dates are unfortunately unknown. Bellow graphs show the growth of the collection over time.

3.3.1.1 *Recordings*

Recordings overall and monthly upload rate over time. Keep in mind that the upload rate is shown on the graph at a different scale based on the axis to the right. Fig. 6 shows a recording from the different continents concerning time constraints.

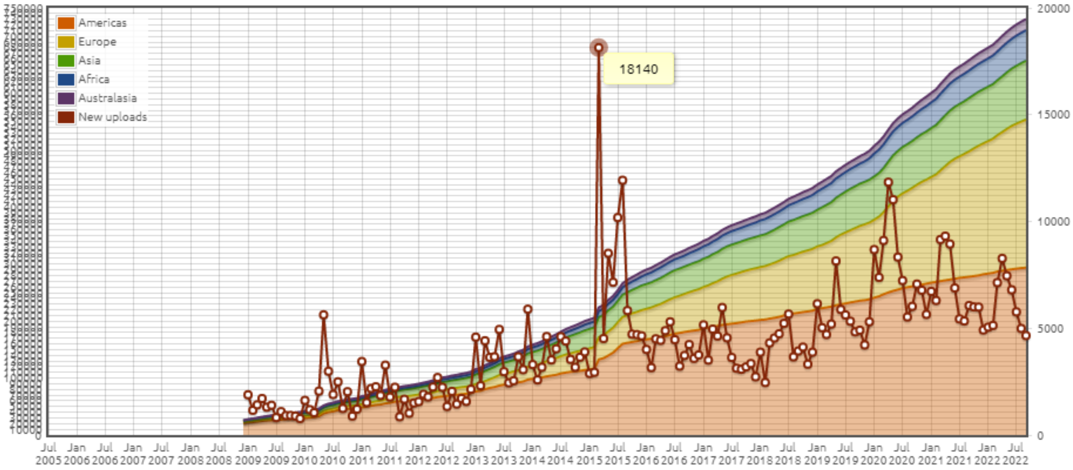

Figure 6. Overall Recordings with time

3.3.1.2 Species

Fig 6 shows the number of species represented by at least one recording in the collection.

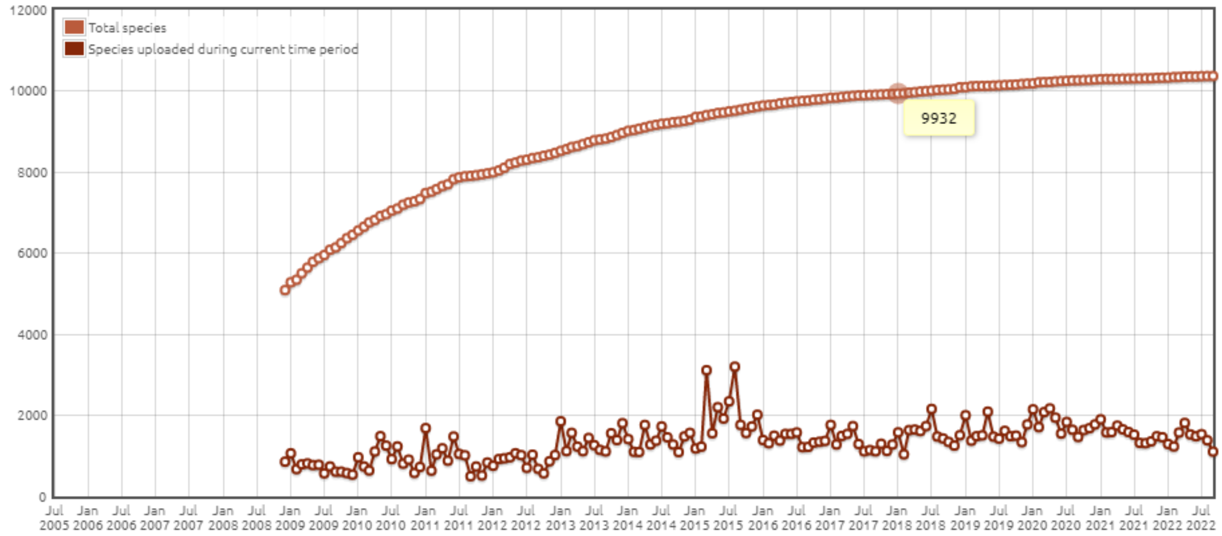

Figure 7. Species Represented by at Least one Recording

3.3.1.3 Contributors

Fig. 8 shows how many users have contributed at least one recording to the collection.

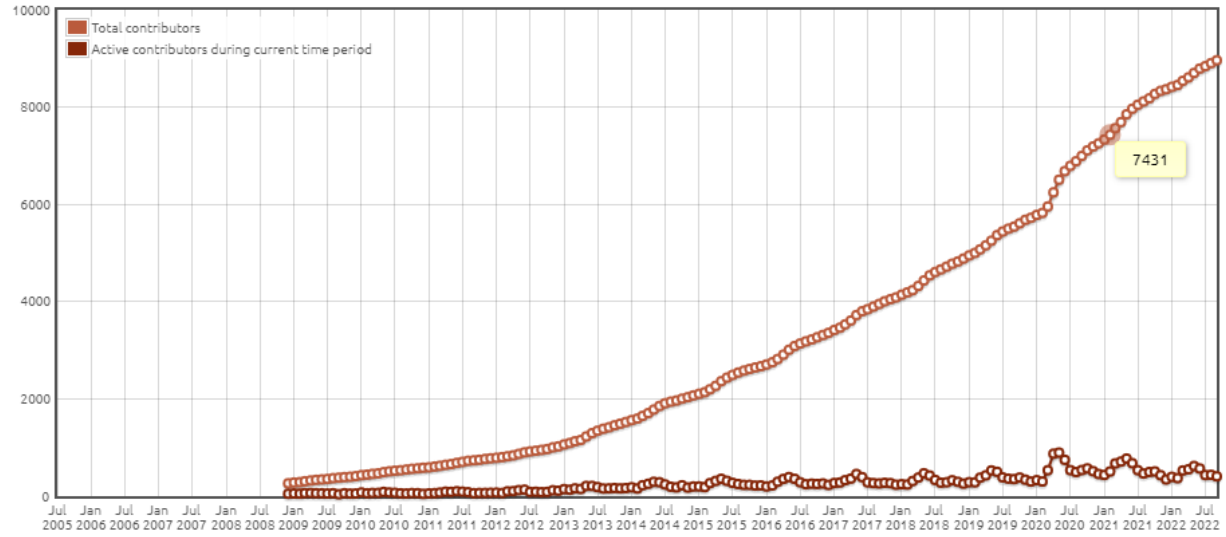

Figure 6. The user that Contributed at least one Recording

3.3.1.4 Recordists and Species vs. Recordings

The number of recordists and species is a function of the number of recordings instead of time. Fig. 9 shows the recordist and species with respect to recording frequency.

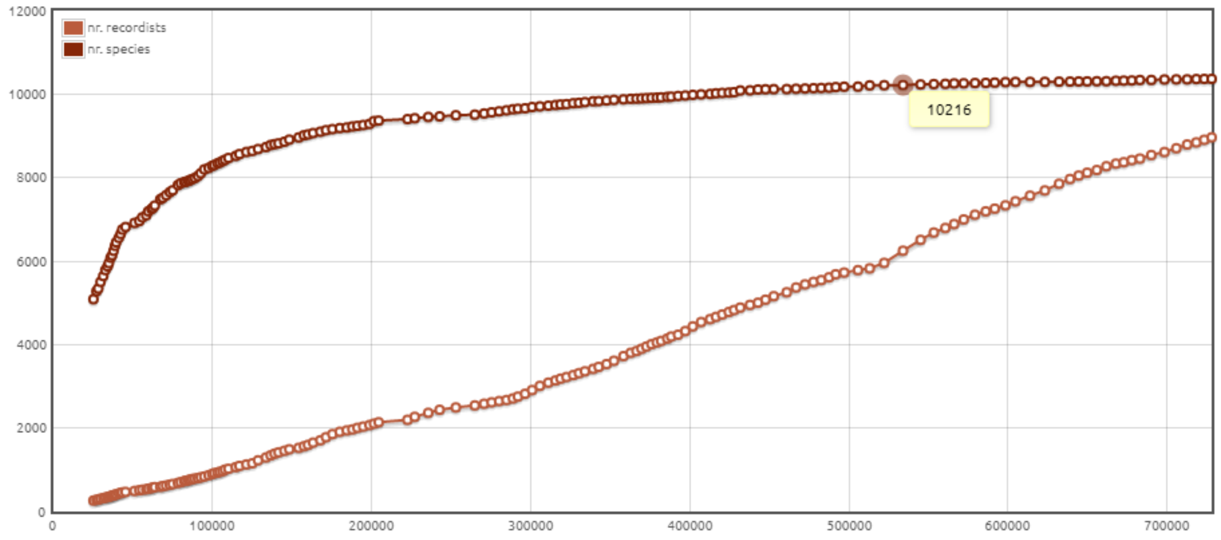

Figure 7, Number of Recordists and Species

3.3.1.5 Recording Year

Several recordings and species as a function of the year they were recorded. Fig. 10 shows the number of recordings and number of species from 1980 to the 2022 year.

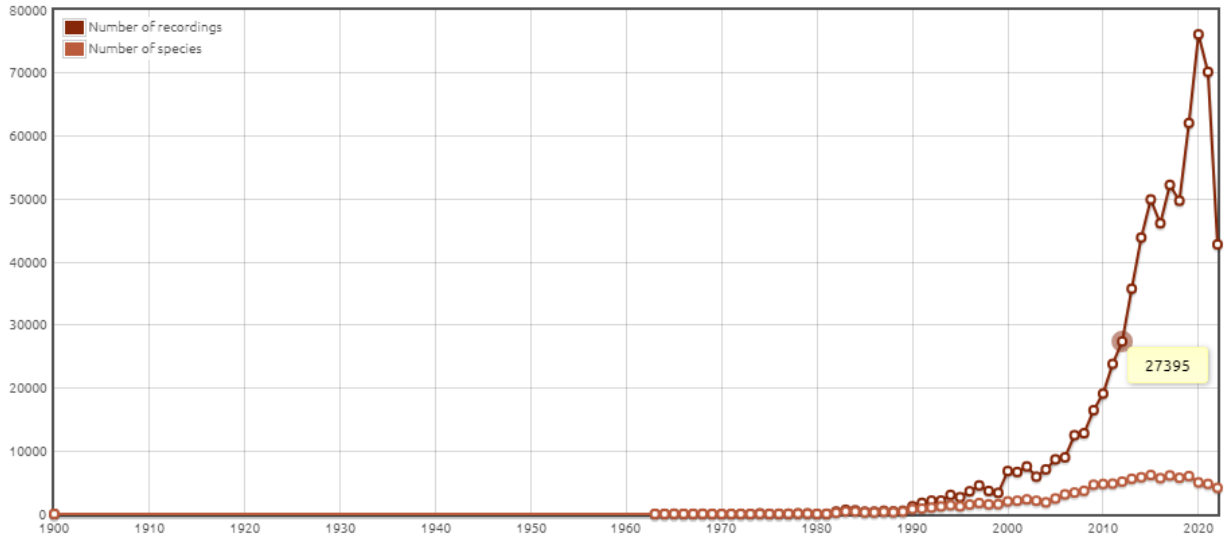

Figure 8. Recordings and Species concerning years

3.3.2 Recordist Statistics

A list of all recordists and their contributions to the sound of the bird. The table shows the top ten names of all recordists, sorted by the number of recordings. Table 2 shows the statistics about the recordist with different parameters and Fig. 11 shows the graphical representation of the statistics of the recordist.

Recordings: the number of recordings by that recordist

Species: the number of species by that recordist

Unique: the number of species contributed only by that recordist ('xeno-canto recordist endemics')

Duration: the total duration of recordings contributed by that recordist

Table 1. Top Recordist Statics with recordings, species, unique and duration

Recordist Name	Recordings	Species	Unique	Duration
Peter Boesman	34059	5089	31	297:39:16
david m	28609	177	2	1093:48:50
Frank Lambert	18584	4559	67	225:28:35
Stanislas Wroza	13202	489	0	258:28:26
Albert Lastukhin	13175	1114	2	206:38:14
Richard E. Webster	12499	744	0	395:08:44
Mike Nelson	11919	2634	12	116:26:33

Niels Krabbe	10994	2016	12	134:26:12
Andrew Spencer	9368	3126	31	124:58:56
José Carlos Sires	9022	216	0	831:54:13

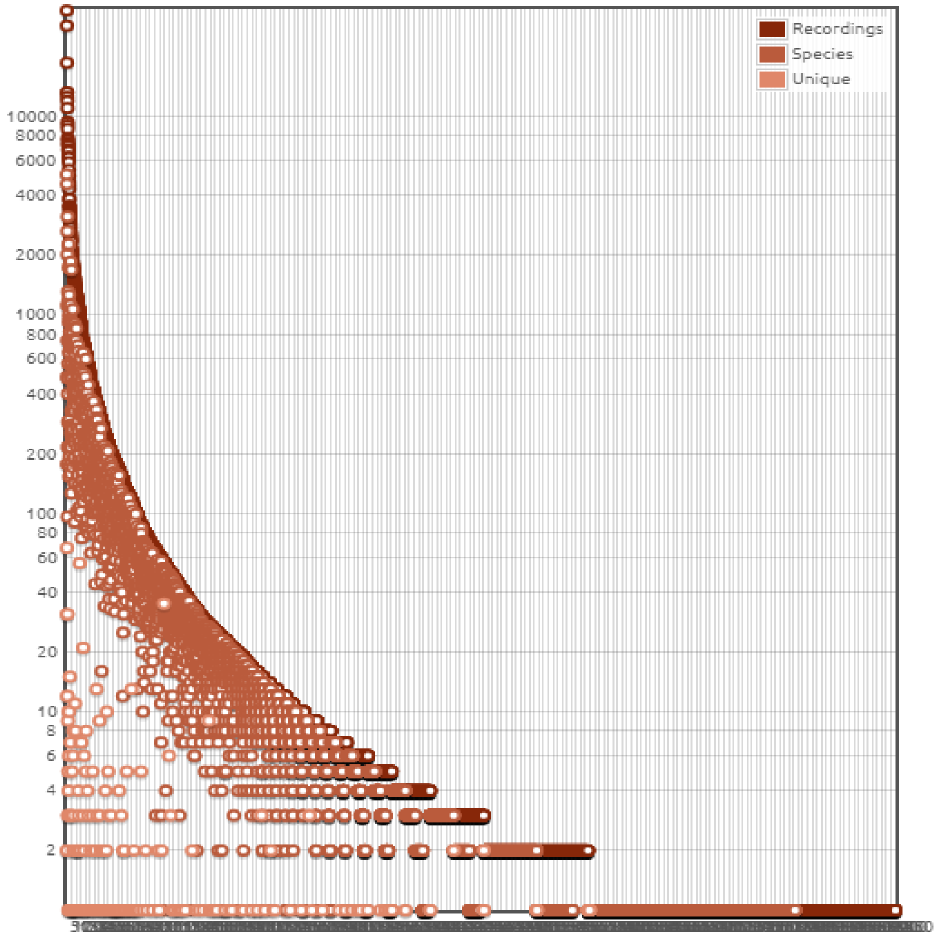

Figure 9. Number of species by that Recordist

3.4 Species

The species contains three types based on the classes.

3.4.1 All Species

A list of all of the species represented in the collection. A list of all of the species represented in the collection.

3.4.2 Wanted Species

A list of species that are still missing from the collection, by country e.g., we just grab the wanted species of the United Kingdom which shows us that 657 total species in the checklist for the United Kingdom. Note that this list may contain rare or vagrant species.

3.4.3 Latest New Species

A list of the latest species that were added to the collection. It is a list of the most recently added bird species in the xeno-canto collection, in reverse order of appearance.

3.5 World Areas

A dashboard summarizing the status of the collection for each major world area

3.5.1 Africa

Africa: Collection Details:

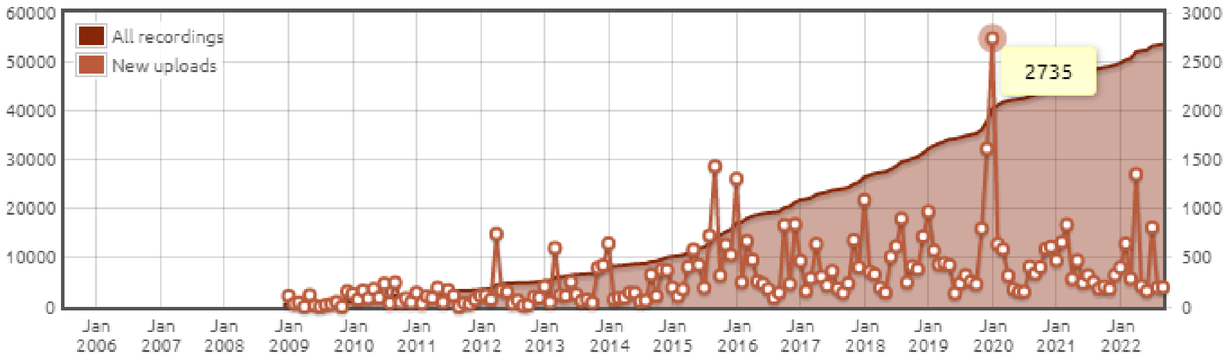

Figure 10. Recording Collected from Africa Continent over the time

Collection Statistics:

Table 2. Recording Statistics from Africa Continent

Recordings	53507
Species	2132
Recordists	582
Locations	4983
Countries	64
Recording Time	573:46:53

3.5.2 Americas

Americas: Collection Details:

Fig. 13 shows recordings collected from the Americas continent since 2006.

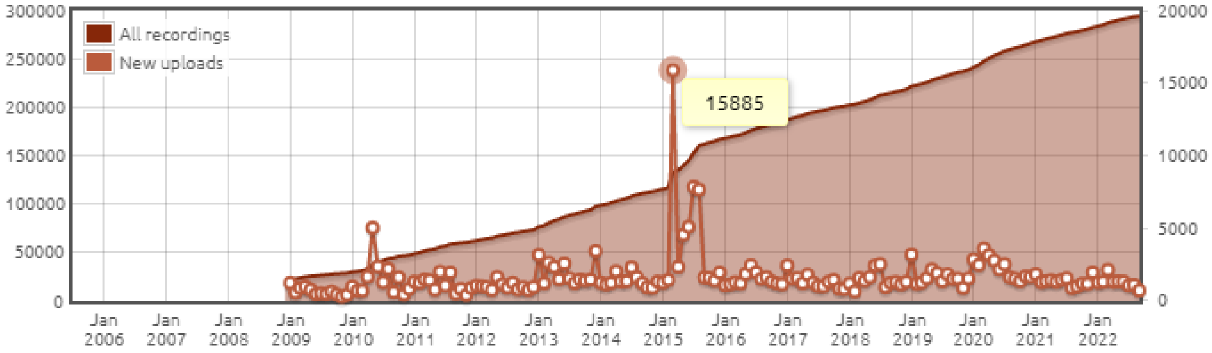

Figure 11. Recording Collected from the Americas Continent over the time

Collection Statistics:

Table 3. Recording Statistics Collected from the Americas Continent over the time

Recordings	293897
Species	4445
Recordists	3450
Locations	31279
Countries	44
Recording Time	3850:36:55

3.5.3 Asia

Asia: Collection Details:

Fig. 14 shows recordings collected from the Asia continent since 2006.

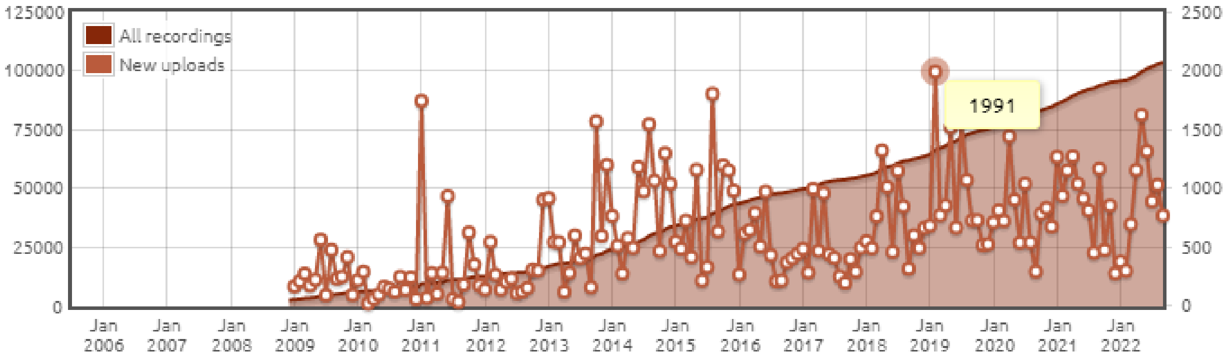

Figure 12. Recording Collected from Asia Continent over the time

Collection Statistics:

Table 4. Recording Statistics Collected from Asia Continent over the time

Recordings	103388
Species	3445
Recordists	1617
Locations	10751
Countries	55
Recording Time	1234:22:26

3.5.4 Australasia

Australasia: Collection Details

Fig. 14 shows recordings collected from the Australasia continent since 2006.

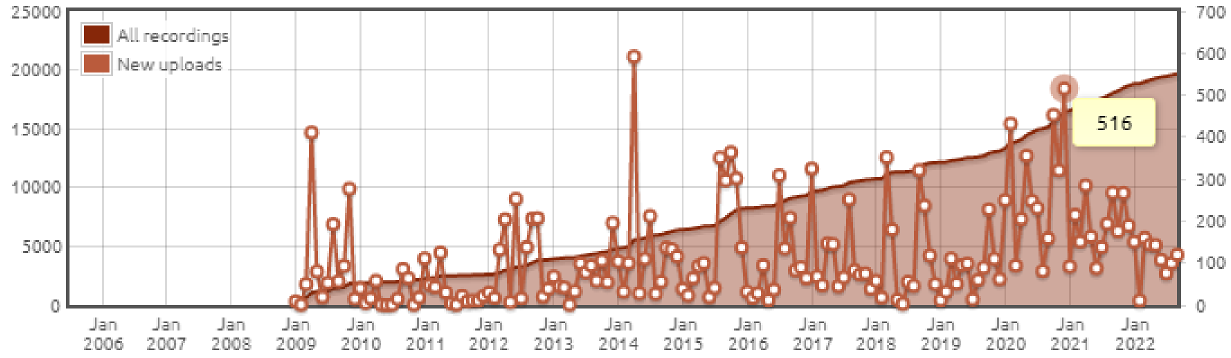

Figure 13. Recording Collected from Australasia Continent over the time

Collection Statistics:

Table 5. Recording Statistics Collected from Australasia Continent over the time

Recordings	19679
Species	1451
Recordists	394
Locations	2711
Countries	17
Recording Time	240:52:45

3.5.5 Europe

Europe: Collection Details:

Fig. 16 shows recordings collected from the Europe continent since 2006.

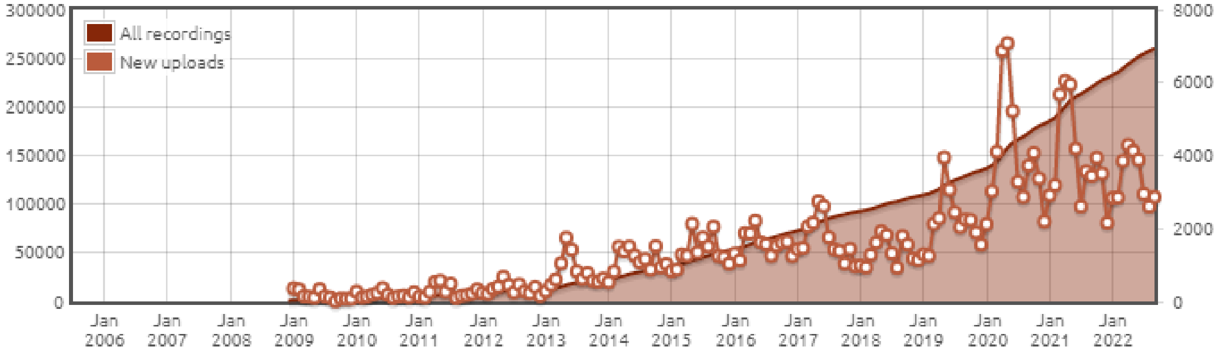

Figure 14. Recording Collected from Euroup Continent over the time

Collection Statistics:

Table 6. Recording Statistics Collected from Euroup Continent over the time

Recordings	259978
Species	753
Recordists	4094
Locations	28178
Countries	52
Recording Time	5986:11:58

3.6 Meta Data Description

In nature, birds perform a crucial function. They are at the top of the food chain and take into account changes at lower echelons. Birds are so great indicators of both environmental contamination and declining habitat quality. But it is frequently simpler to hear birds than to see them. There are more than 10,000 different bird species around the globe, and they may be found in almost every setting, from pristine rainforests to suburbs and even cities. Researchers might automatically infer information about an area's quality of life-based on a shifting bird population with the right sound detection and identification techniques.

Several initiatives are already underway to monitor birds in great detail via the long-term, continuous recording of natural soundscapes. However, because so many live and nonliving entities produce noise, domain specialists frequently analyze these datasets manually. These analyses are laboriously slow, and the outcomes are frequently not comprehensive. Researchers have used sizable, crowdsourced datasets of bird recordings to train AI models because they believe data science may be able to help. Unfortunately, there is a domain incompatibility between the soundscape recordings utilized in monitoring applications (long recordings with frequently numerous species calling at the same time) and the training data (brief recordings of individual birds). This is one of the factors contributing to the poor performance of the AI models that are now in use. Researchers require effective machine listeners to reliably extract as much information as possible to support data-driven conservation to fully use the promise of these huge and information-rich sound archives.

The dataset exists in two different modalities: 1) Meta Data which contains different varieties of details about the bird e.g., rating playback_used, ebird_code, channels, date, pitch, duration, filename, speed, species, xc_id, url, country, author, primary_label, latitude, longitude, length, time, recordist, license. The initial data shape is in JSON format. In the data engineering and preprocessing pipeline, we should convert the data into a pandas data frame to apply processing and data visualizing techniques. So, we changed the dataset into the CSV (comma-separated values) format. This dataset has 46565 rows and 26 columns. The table contains information about these columns and their details.

Table 7. Detailed Description of Meta Data for Xento-Canto Dataset

Sr.	Column	Type	Count -Non-Null	Description
1	id	Int64	46565 non-null	The catalog number of the recording on xeno-canto
2	gen	Object	46565 non-null	The generic name of the species
3	sp	Object	46565 non-null	The specific name (epithet) of the species
4	ssp	Object	46565 non-null	The subspecies name (subspecific epithet)
5	en	Object	46565 non-null	The English name of the species
6	rec	Object	46565 non-null	The name of the recordist
7	cnt	Object	46565 non-null	The country where the recording was made
8	loc	Object	46567 non-null	The name of the locality
9	lat	Object	46567 non-null	The latitude of the recording in decimal coordinates
10	lng	Object	46565 non-null	The longitude of the recording in decimal coordinates
11	alt	Object	46565 non-null	Alternative text
12	type	Object	46565 non-null	The sound type of the recording (e.g. 'call', 'song', etc.). This is generally a comma-separated list of sound types.
13	url	Object	46565 non-null	The URL specifying the details of this recording
14	file	Object	46565 non-null	The URL to the audio file
15	file-name	Object	46565 non-null	The original file name of the audio file
16	sono	Object	46565 non-null	An object with the urls to the four versions of sonograms
17	lic	Object	46565 non-null	The URL describing the license of this recording

18	q	Object	46565 non-null	The current quality rating for the recording
19	length	Object	46565 non-null	The length of the recording in minutes
20	time	Object	46565 non-null	The time of day that the recording was made
21	date	Object	46565 non-null	The date that the recording was made
22	uploaded	Object	46565 non-null	The date that the recording was uploaded to xeno-canto
23	also	Object	46565 non-null	An array with the identified background species in the recording
24	rmk	Object	46565 non-null	additional remarks by the recordist
25	bird-seen	Object	46565 non-null	Was the recorded bird visually identified? (yes/no)
26	playback-used	Object	46565 non-null	Was playback used to lure the bird? (yes/no)

3.7 Bird’s Voice Data Description

The dataset we gather is the subset of the xento-canto database. The goal of the Center for Conservation Bioacoustics (CCB) at the Cornell Lab of Ornithology is to gather and analyze sounds from the natural world. The CCB creates cutting-edge conservation technology to motivate and educate people throughout the world about the need to save species and environments. The CCB seeks to further its goal and enhance the precision of soundscape assessments by collaborating with the data science community. We need to execute data processing and data engineering to choose the optimal label for the bird's sound in soundscape recordings where there are several different bird vocalizations. Weak labels are present in the recordings because of their intricacy. A specifically named bird species may be in the forefront, with anthropogenic noises (such as airline overflights) or other bird and non-bird calls (such as chipmunk cries) in the background.

3.7.1 Dataset Folders Structure Details

Train audio:

The training data is made up of shorter recordings of certain bird cries that xenocanto.org users have kindly shared.

Test audio:

About 150 mp3 recordings, each lasting about 10 minutes, are concealed in the test audio directory. They won't all fit in the memory of a laptop at once. Three different isolated sites in North America are where the recordings were made. Sites 1 and 2 were labeled at 5-second intervals and need matching predictions, while site 3 files are only labeled at the file level due to the time-consuming labeling procedure. As a result, site 3 has a test set with fewer rows and requires predictions with lower temporal resolution.

- site: Site ID.
- row_id: ID code for the row.
- seconds: The second ending the time window, if any. Site 3-time windows cover the entire audio file and have null entries for seconds.
- audio_id: ID code for the audio file.

Example test audio metadata.csv

Complete information for the test audio example. Compared to the concealed test set, these labels are more time precise.

Example test audio summary.csv

Metadata for the sample test audio was converted to the concealed test set's format.

- filename_seconds: a row identifier.
- birds: all bird codes are present in the time window.
- filename
- seconds: the second ending the time window.

train.csv

The training data comes with a wide variety of information. The fields that are most immediately relevant are:

bird code: The bird species' code is bird code. By adding the code to https://ebird.org/species/, such as https://ebird.org/species/amecro for the American Crow, you may access full information on the bird codes.
recordist: the user who submitted the recording.
location: the site of the recording. You might wish to look for geographic variability in your training data since certain bird species may have regional cry "dialects".

date: While certain bird cries, like an alarm call, can be heard all year long, others are only made during a certain season. It could be a good idea to look for temporal variability in your training data.

filename: The name of the audio file that goes with it.

3.8 Spectrogram Data Description

The frequency range between 1.8 and 1.9 kHz was most frequently utilized by woodland birds (16% of all motif components), whereas the range between 2.2 and 2.3 kHz was most frequently used by city species. Male forest birds were substantially more frequent users of frequencies below 2 kHz than were city birds. Additionally, Wavelet Denoising simply creates a signal from a noisy one to remove noise from a signal. Wavelet denoising is what I use after the MEL Spectrogram, however, it would be preferable to use it after signal filtering.

The voice data from the above-structured folder is converted to a MAL Spectrogram and then label to classify by using a convolutional neural network. We have performed several experiments to remove any data imbalance issues and data labeling issues during training the model. Figure 17 better comprehend; we display the MEL Spectrogram feature maps for representative signals. The MEL Spectrogram, as can be seen, provides visual data regarding the frequency and amplitude patterns in the audio signal over time.

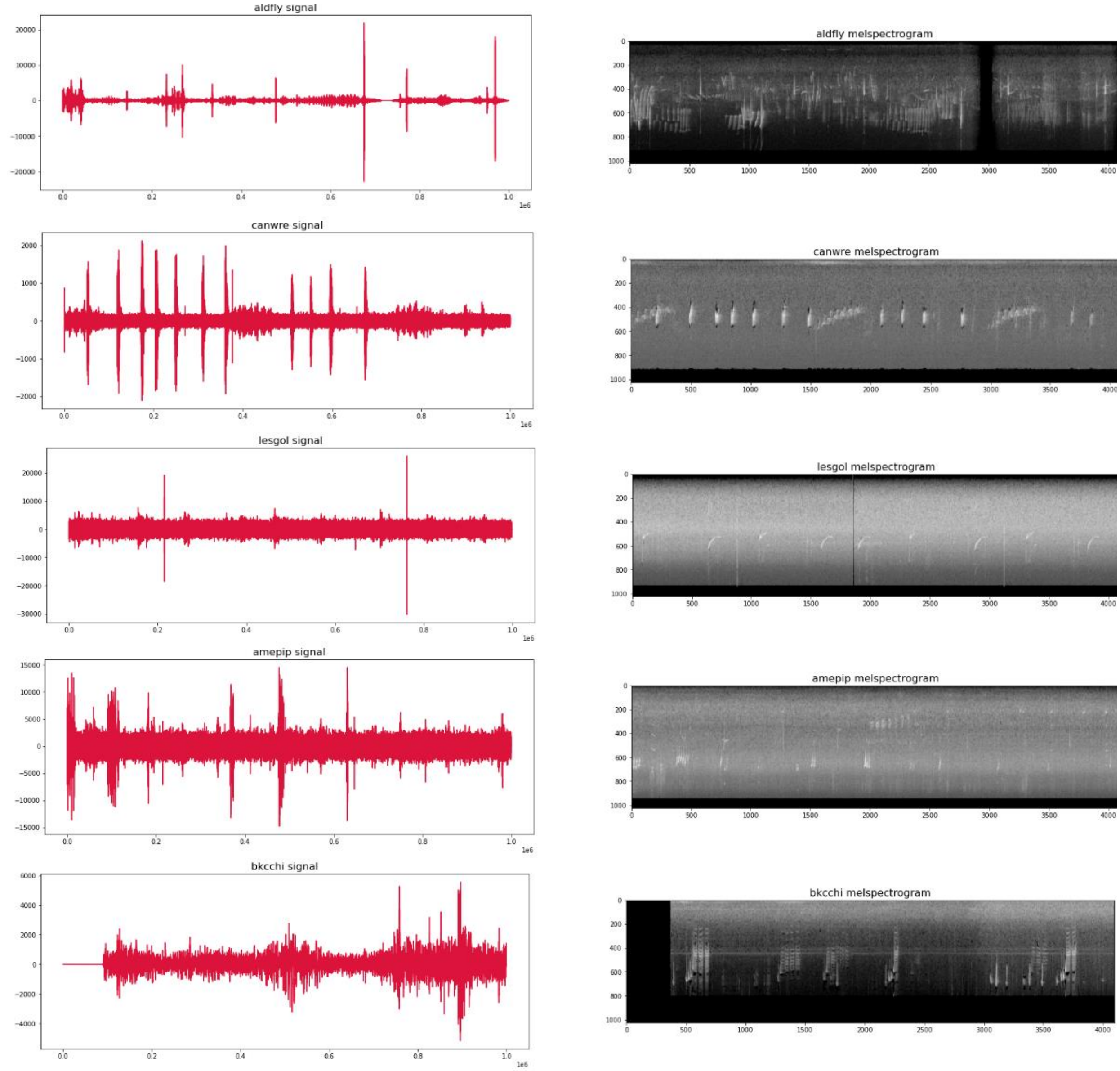

Figure 15. Sound Signal with MEL Spectrogram of 5 species of xento-canto Dataset.

The Fourier transform must first be understood to comprehend the MEL Spectrogram. Simply said, it is a transformation that enables efficient analysis of the frequency content of a signal. But regrettably, for the Fourier transform to work best, the signal must be periodic. The Fourier transform may be used to overcome this issue by capturing the frequency content as it varies over time in different windowed parts of the signals. The short-time Fourier transform, or STFT, is what is used for this. A Spectrogram is produced when the Fourier transform is performed to overlapping windowed portions of the signal. Consider a Spectrogram as a collection of Fourier transforms piled on top of one another. It is a technique for displaying the loudness, or amplitude, of a signal as it changes over time and at various frequencies.

The color dimension is transformed to decibels, and the y-axis is changed to a log scale. This is because only a very narrow and focused range of frequencies and amplitudes can be perceived by humans. Additionally, research has revealed that people do not perceive frequencies on a linear scale. Lower frequencies are easier for us to distinguish than higher frequencies. Even if the distance between the two pairs is the same, we can readily distinguish between 500 and 1000 Hz but will find it difficult to distinguish between 10,000 and 10,500 Hz. Stevens, Volkmann, and Newmann developed a unit of the pitch in 1937 so that the listener would perceive equivalent distances in pitch as equal lengths. It is known as the MEL scale.

3.9 Summary

In this chapter, we analyze the dataset along with metadata and image data in the form of a MEL Spectrogram converted from a wav file to a 2D tiff, PNG image file. We will further analyze the metadata for our exploratory data analysis to make more details about the dataset and its metadata.

4 Chapter 4: Data Description and Exploratory Data Analysis

4.1 Introduction

This chapter covers general exploratory data analysis and preprocessing steps of metadata for the bird voice and the voice data for the respective birds, we further implement the methodology and implementation details to detect and classify bird voice detection. Data collection and selection steps are already discussed in the above chapter.

4.2 Exploratory Data Analysis and Cleaning

There are about 264 bird species in the dataset and for each species multiple recordings are present. Analyzing the dataset after exploration reveals that null and missing values are also present. While learning meaningful full information, we must deal with all the outliers and missing variables that affect accuracy. In this data multiple types of garbage and null values like e.g. "n.a.","?","NA","n/a", "na", "--","-","n.a", "am", '[]', "?:?", "['']", "unknown", "??:??". We must eliminate all forms of trash values and swap them out for NaN values before we can use strategies for missing datasets. The data cleansing method is now simple to use. This dataset contains some NaN values, as indicated in Table 2.

Table 8. Missing values details.

Column	Total Null Values	Percentage
ssp	33385	71.6955
also	18453	39.6285
rmk	6775	14.5496
playback-used	2442	5.24428
bird-seen	2015	4.32728

These missing values have been rectified and eliminated. The NaN cells are filled using a variety of methods, including the mean and mode of the columns. There is a lot of information in the date column that has to be separated for further research. We created three additional columns from the date column: day of the month, month, and year. These additional columns make it easier for us to analyze data based on days, months, and years. Except for the id, all of the columns in the dataset were of the object datatype. Since all of the data is in object type, type casting of all columns is necessary to turn it into the appropriate data type for each column's nature. It includes values starting at 0,00 in the year columns. These filthy values were cleaned by using filters.

We analyze the dataset by finding different graphs for each column and group of columns to know about the bird species and how they are propagating through different countries and continents.

4.2.1 Species for North and South England

North England has more bird species residing there than South England does. There is no information regarding the north and south of England in the data. The loc column provides specific geographic information for each bird species. To do this, we filter each location using the North and South keywords in the "loc" col. The top species that are prevalent in the south and north England, respectively, are first identified as illustrated in Fig. 18. Since we now have frequent species in both the north and the south, we can once more filter out "a specie that is more frequent in North England than the south of ENGLAND" by utilizing pandas data frame quires. In the North of England, Trochilus was discovered 2499 times, but just twice in the South. Therefore, we conclude that "Trochilus" and "Mystery" are the most common bird species in North England.

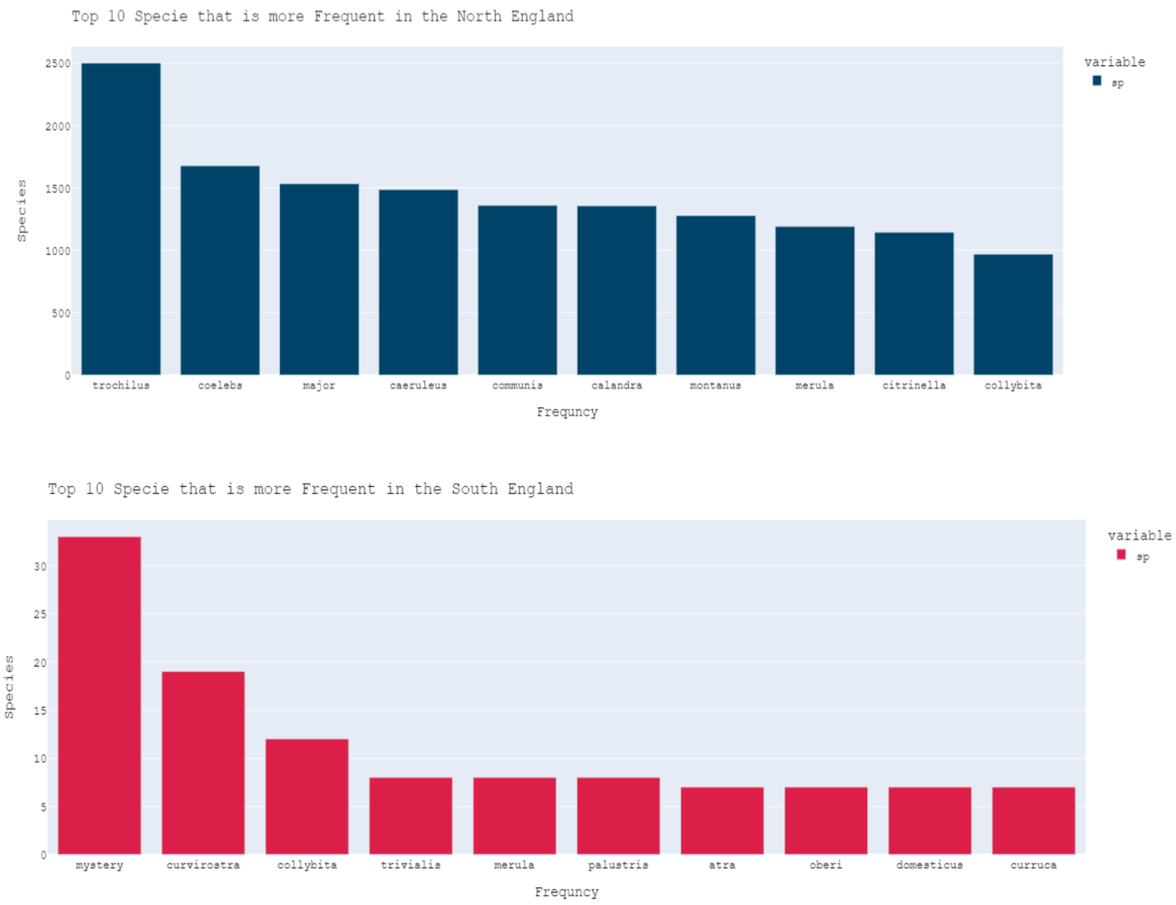

Figure 16. Most Frequent Species in North and South of England

4.2.2 Most Frequent Singing Birds in Morning and Evening

Because birds are such an intriguing creation of GOD, we may examine the data to determine whether any species sing more frequently in the morning than in the evening. To turn the time, we have in minutes and seconds into an hour, we must first convert it into hourly data. To retrieve the data's hour, we first constructed an hour column using the ":" character. Currently, we have an hour column but no session of the day, such as "Late Night," "Early Morning," "Morning," "Noon," "Evening," or "Night." Hours must be converted into various daily sessions before being converted into these sessions. We convert it on an hourly basis so that the 24 hours in a day are divided into pieces and mapped into their corresponding sessions. The top 50 species that sing more frequently in the morning and evening are obtained by performing the query, as illustrated in Fig. 19. We created a news column titled session by hourly filtering the data based on morning and evening time. We examine the column in light of the species that were more prevalent in the morning and evening. Bird Collybita appeared 1158 times in the morning and 36 times in the evening, as seen in Fig. 4. Collybita is therefore more usually sung in the morning than in the evening, according to the outcome of this scenario.

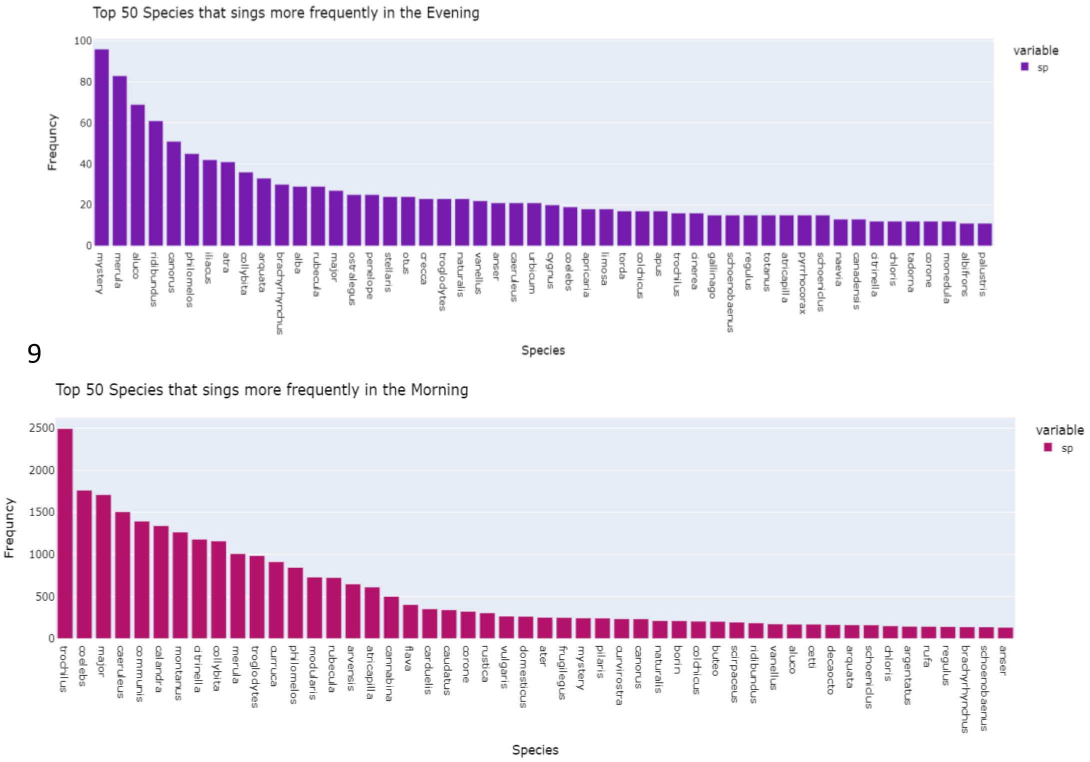

Figure 17. Top 50 Species that sing more frequently in the Morning and Evening

4.2.3 Geographical representation of Birds

Birds travel as the weather changes over time, allowing us to study the position of the birds and their geographic distribution to determine when the majority of them are present throughout the year and when they sing. "Work on the geographical distribution of the bird species" is required of us. First, you need to learn where the recordings were produced. I'll be plotting the precise places where the recordings were made on the map as I know the latitude and longitude of the spot where they were produced. Consider the top 15 nations with the most recordings. The UK has the most records, followed by the cities there. To plot country/city names on a map, the first step is to convert them to iso-alpha format. We must remove the nation name from the location column because it contains highly noisy data. Thus, we used "," to divide the values in the "loc" column. After that, we extract the nation's name and store it in the new country column because the country name is present after "," Following the development of the country column, we must clear up any countries with confusing data. We then update the countries concerning their cities after personally checking them and changing their names to reflect their states. The name of the main nation, which is the United Kingdom, is converted into short or iso-alpha code before being plotted on a map. This format is known as a country short form or iso-alpha. Since there is no other nation mentioned in the latitude and longitude, we may infer from the data that it is from the United Kingdom. As a result, we assume that we won't be adding another database for the bird to this data due to inconsistency. Normal Distribution without a Map, and as we can see, the majority of the birds are from the UK, with just a small minority coming from other continents or nations. The distribution of the England Map and locating the spices are depicted in Fig. 20.

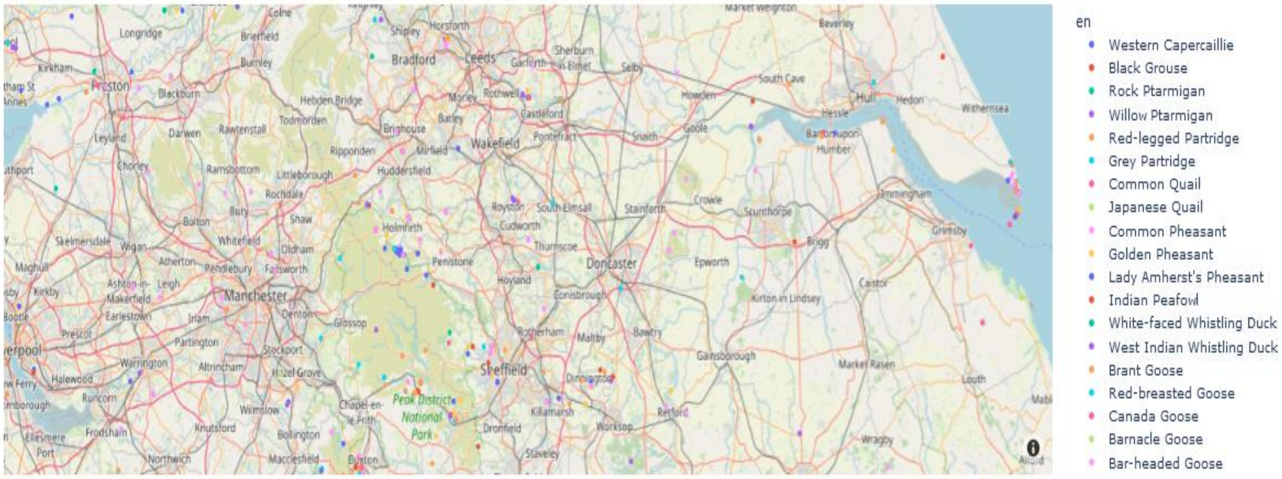

Figure 18. Distribution of Map of England and mapping the location of the spices

4.2.4 Most Bird Recordings

The top 15 altitudes for the bird's species according to the number of occurrences in the newly added nation column. The top 15 bird species are shown in Fig. 21, and the percentage of the most common bird species is shown in Fig. 22.

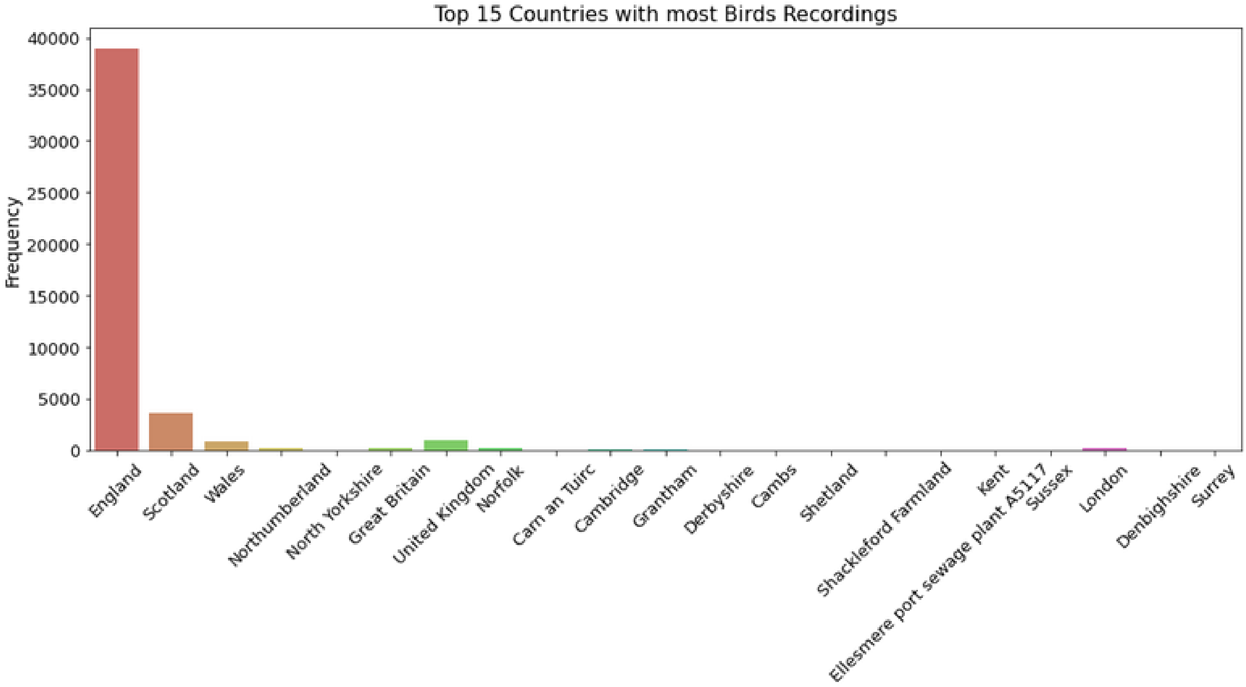

Figure 19. Top 15 Country having Most Bird Species Recording

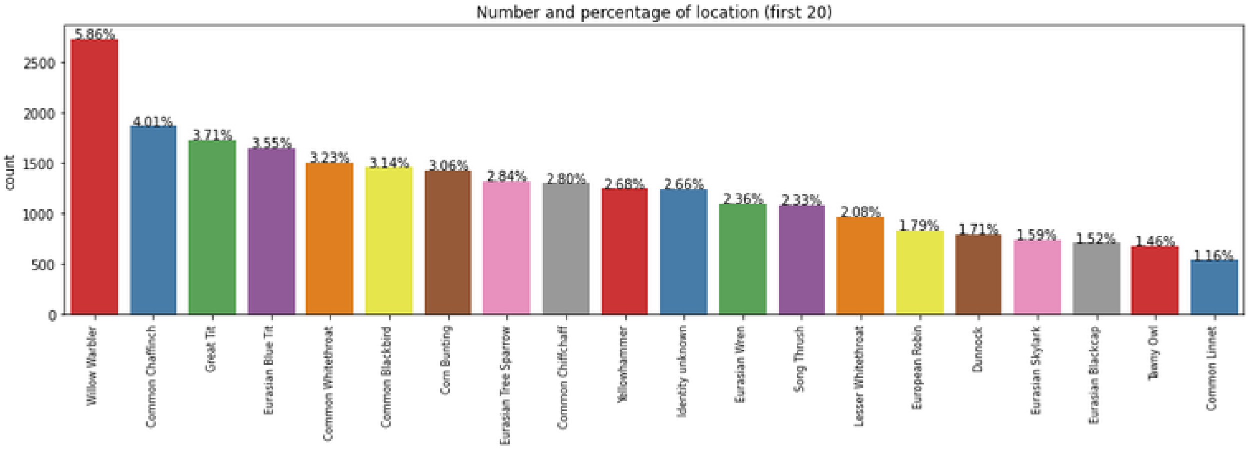

Figure 20. Most Occurred Bird's Species Percentage

We determine that the most frequent bird recordings are from England, as shown in the Geographical Distribution, and that the most frequent bird recordings are from the United Kingdom specifically from England, based on the aforementioned operations of data cleaning and data grouping/aggregation based on another significant factor. By performing additional analyses, we identify the English city of "Willow Warbler" as the one with the highest bird population.

4.2.5 Frequent Uploader in Session

We must analyze the time of day and even choose a different season, such as winter. Do we record more in the spring than in the winter? Since we modified the date column, the data include months. The distinct seasons of the year must now be located. Therefore, we must translate the months into their corresponding sessions, such as "Winter," "Spring," "Summer," and "Autumn." The session is double-checked to the weather/session in England. Identifies the bird species recordings that were uploaded the most often during the spring session, as seen in Fig. 23.

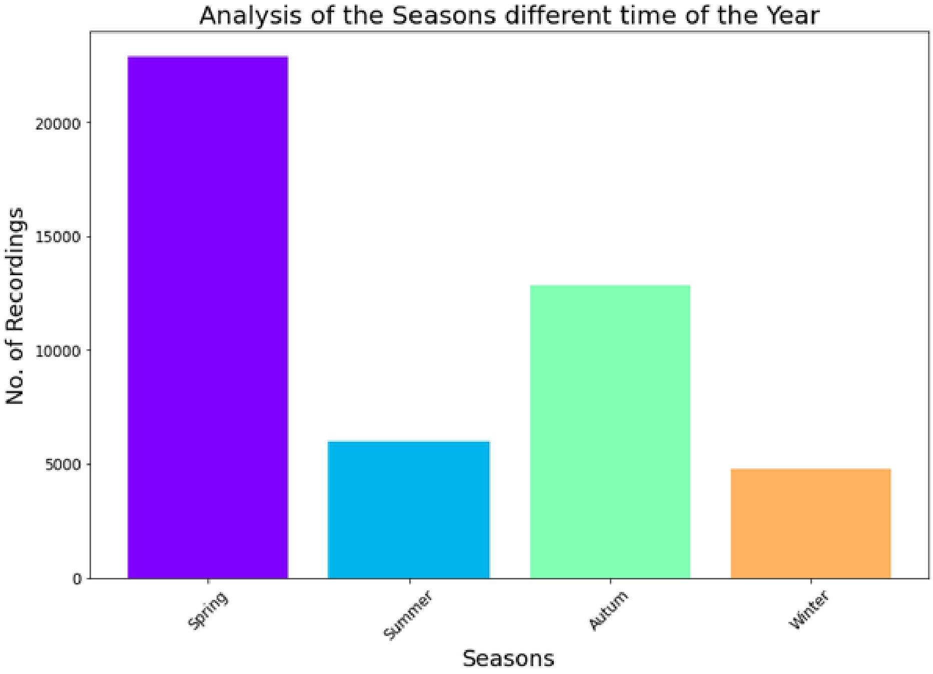

Figure 21. Most Frequent Recording in Different Seasons of the Year

4.2.6 Bird Year Analysis WRT Years and Months

Now analysis of bird recording concerning Years. Finds the most frequent uploading of the bird species recordings in year columns which are 2020 as shown in Fig. 24.

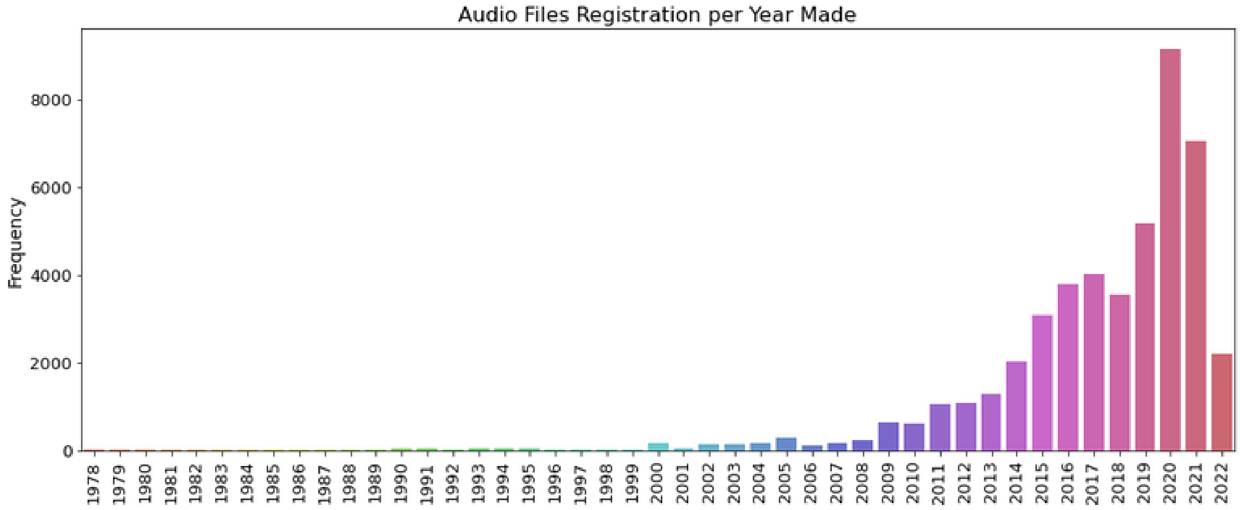

Figure 22. Frequent Recording with respect to Years

We also conducted a month-by-month study of bird recordings. based on months, determines which bird species' recordings are uploaded the most frequently. Therefore, as shown in Fig. 10, we determine that the soring session and the month of April are more appropriate for documenting bird species.

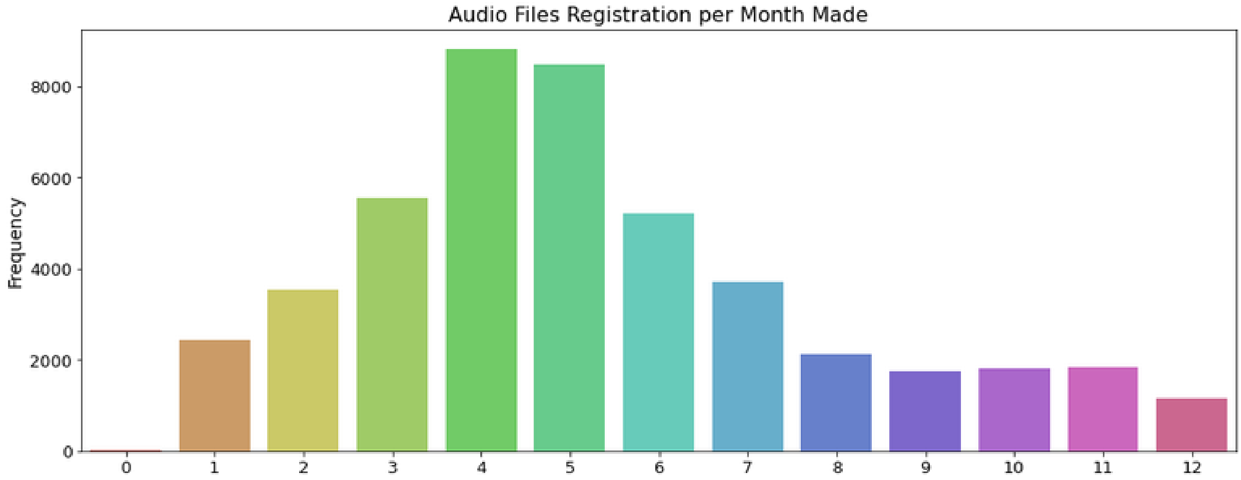

Figure 23. Most Frequent Recording w.e.t concerning Months

According to the recorded voice frequencies for each month, as seen in the graph, we have more recordings in the spring than in the winter. April is the most particular month when Recorder registers 8000+ Audio files. Thus, we conclude that Audio file registration may be considering the time of year's spring in the early morning hours. The UK's current weather is as follows: A period of unexpected rain showers, blossoming trees, and flowering plants is spring (March, April, and May). The hottest months in the UK are summer (June, July, and August), which has long, sunny days, sporadic thunderstorms, and, on occasion, heatwaves. The autumn months of September, October, and November can be either pleasant and dry or windy and damp.

4.2.7 Most Recording Uploads

We also analyze the most frequent voice recording uploader and how much he is uploading during his career and what is his behavior of the person based on data upload. We have previously discovered that audio recording registration is frequently done, therefore we will examine this query in light of the Uploader Person. We will start by locating the regular uploader of morning and evening bird species audios. The next step is to determine the uploader's activity, including the year, month, and day of the month they uploaded the tape. In this research, we discover that David M. uploaded the recording the most, 27776 times, as shown in Fig. 26. As illustrated in Fig. 27, the Uploader may upload more often at night or in the morning, or at different times of the day or year.

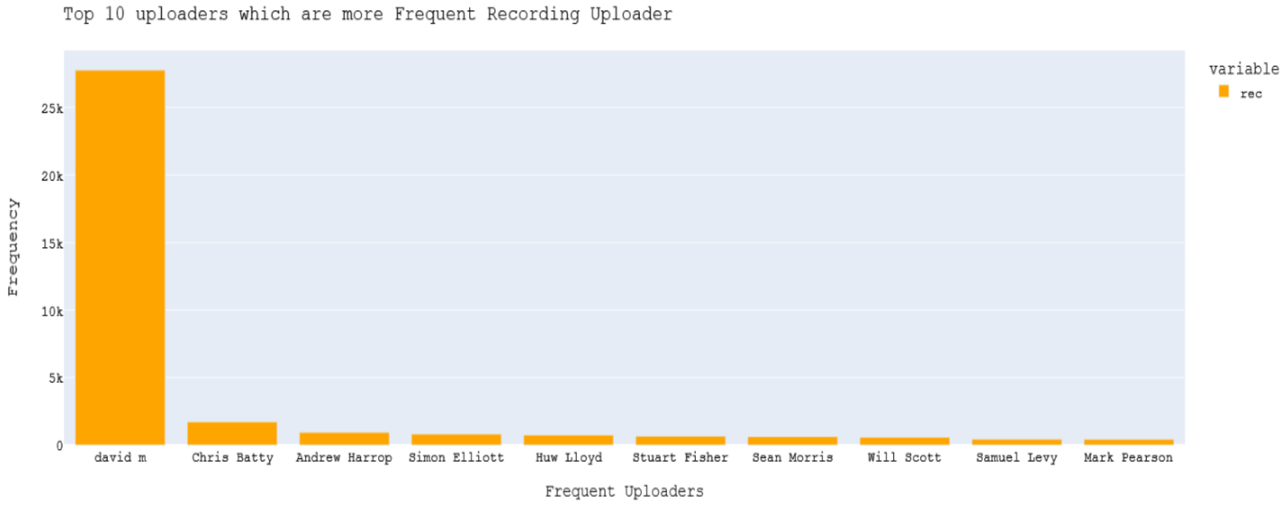

Figure 24.Top 10 Uploaders Which Are More Frequent Recording Uploaders

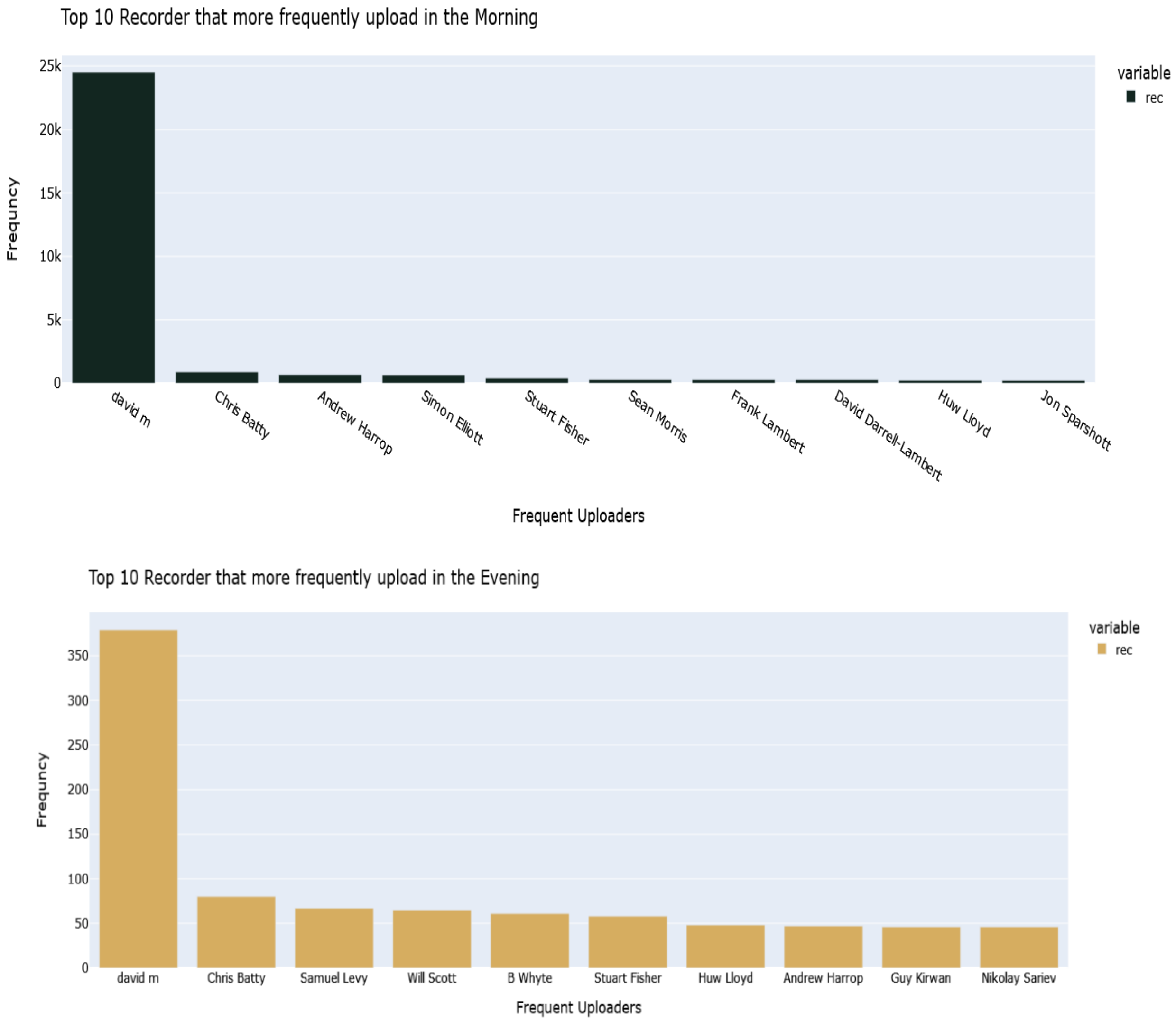

Figure 25.Recorder That More Frequently Upload in the Evening and Morning

The metadata contains the ratings of songs in the range of 0.0 up to 5.0. The 5 rating songs are mostly clear and it's the best song the recorder recorded. In Fig. 28 we are analyzing how many songs are belonging to 0 to 5.

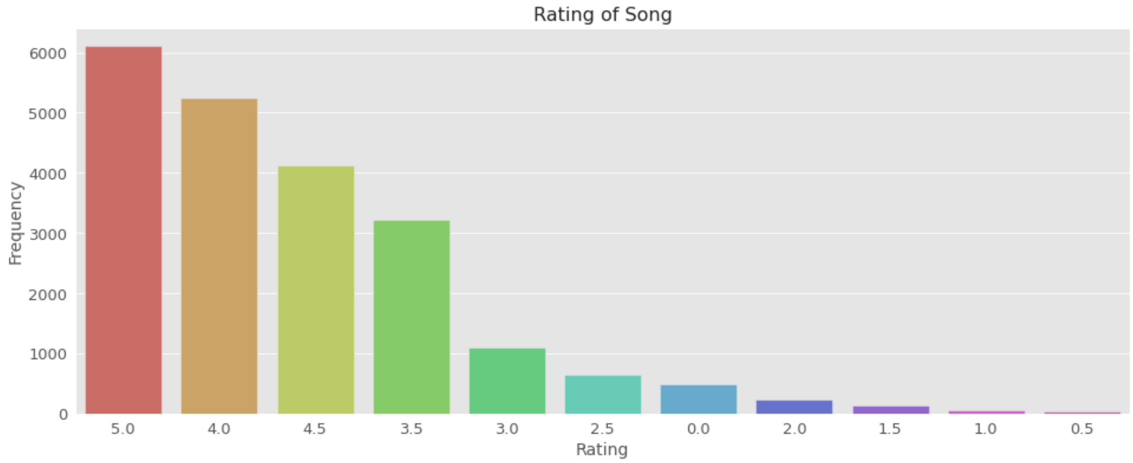

Figure 26. Most Frequent Rating based on occurrence frequency.

The rating distribution concerning frequencies is shown in the graph above, and it is evident that the top-rated bird songs have frequencies of around 6000. Similar to this, we may see a trend where better ratings correspond to increasing frequency and vice versa. Similar to the above graphs we also visualize some of the most occurred bird songs.

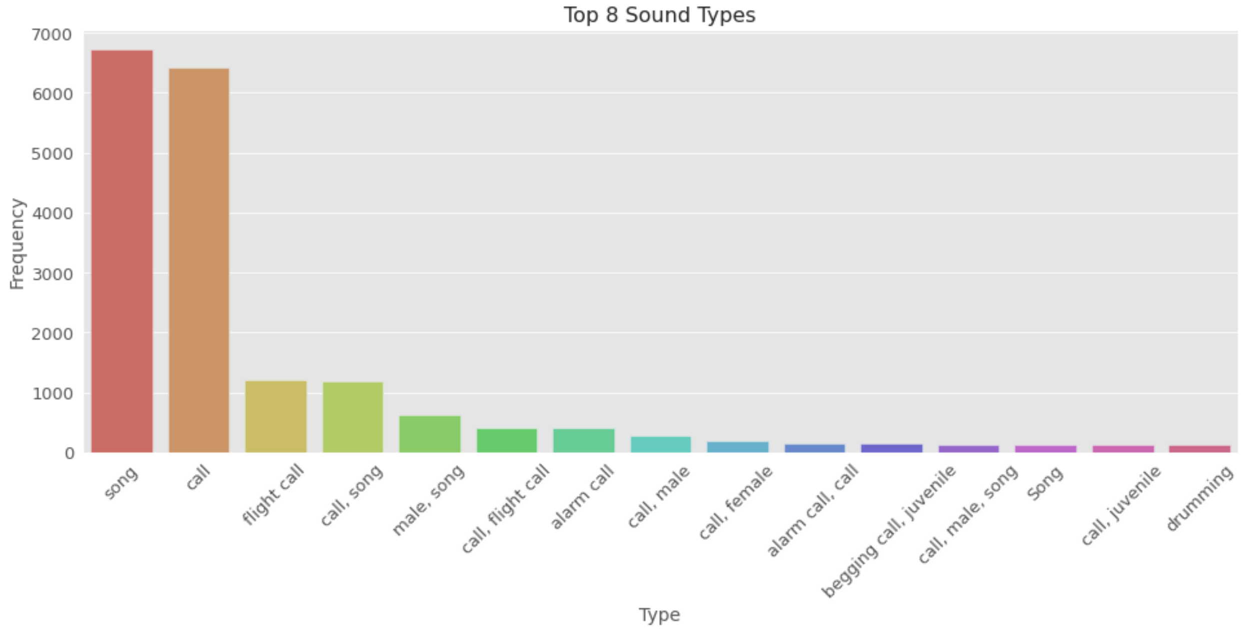

Figure 27. Top sound types and their Frequencies

4.3 Summary

Exploratory data analysis is essential during data analysis for many reasons. Any piece of data is susceptible to mistakes being made during data collection or input. Outliers are among the best indications of data mistakes, and EDA may be used to find these flaws. Box plots may be used to locate outliers in a data collection. Since this type of exploratory data analysis is visual, it is simple to spot numbers in anomalous data. Without doing data analysis, it is hard to determine some aspects during data collection, such as distribution patterns. Our target is to classify different classes of bird species on the image the experimental analysis will be performed below chapter.

5 Chapter: 5 Pre-Processing, Methodology, and Implementation

5.1 Introduction

Deep learning is a type of machine learning inspired by the structure of the human brain. In terms of deep learning, this structure is an artificial neural network. On other hand. On the other hand, Deep learning is a subset of machine learning which in turn is a subset of artificial intelligence. it uses a programmable neural network that enables machines to make accurate decisions without help from a human. Shortly and simply, deep learning is what powers the most human-like artificial intelligence. Just like we use our brains to identify patterns and classify various types of information, deep learning algorithms can be taught to accomplish the same tasks for machines. Object Detection is one of the most demanding and helping concepts in computer vision. Nowadays, it is being used almost everywhere. The most popular deep learning concept is Convolution Neural Network (CNN). In CNN work has been done in layers. There were multiple layers in it. Convolution Neural Network will be used to classify our specified object. Different layers will perform the calculation and every layer will pass their output which acts as input for a new one.

The major reason for using CNN as its precise features without the need for human help. CNN has different types of hidden layers. Each layer has a filter to detect the specific part of the image. Convolution will be applied to the image data by using the filters of convolution to get the feature of the interesting object. Sliding the filter over the input image to perform the convolution operation. Matrix multiplication will be done at every location afterward to sum the results. This result will be going to the feature map. The area where the convolution is applied is called as receptive field. Different convolutions will be applied in the input and will get a diverse feature map. All these features map will b combined to make the final output. This will not be the sum actually as it will be the Relu function applied to them. After that pooling will be preform to reduce the dimensionality and the number of parameters as well. A fully connected layer will wrap up the CNN process. Results of convolution and pooling will be fed to it then it will drive the final results.

5.1.1 Region-Based CNN (R-CNN)

R-CNN was proposed in 2013 by Girshick et al, to overcome the problem of a large number of regions selected by a simple CNN. This method selects only 2000 region proposals for CNN to work on. These regions are selected by a selection search algorithm. Now instead of a large number of possible regions created by a simple CNN, now only 2000 regions will be sent to CNN which extracts the features from those regions and send that output to SVM for classification of the presence of an object and creates a bounding box around it. As R-CNN was the first practical method for image detection, it had some drawbacks such as:

- Training contains multiple stages (i.e., selection of regions, feature extraction, and then classification) so this method was still slow and expensive.
- It cannot be implemented in real-time because it takes almost a minute to process one image.
- The selection search algorithm is fixed so no learning is happening which can lead to generating bad region proposals.

All these issues were resolved in the improved versions of R-CNN such as Fast R-CNN, and Faster R-CNN.

5.1.2 Fast R-CNN

Fast R-CNN was proposed in 2015 by Girshick et al, as the improved version of R-CNN. This shortcoming of R-CNN was removed in this version. Its working is the same as R-CNN but in its place of generating region proposals and then feeding them into CNN, in Fast R-CNN the image is directly fed into CNN to generate a feature map from which the region proposals were selected (using selection search) and then after Rol pooling, a fully connected layer (a SoftMax layer and a real-valued layer) is used to predict the class of the detected object and create a bounding box around it respectively. The entire network (including the Roi pooling layer and the fully connected layers) can be trained using the back-propagation algorithm and stochastic gradient descent.

What makes the Fast R-CNN faster than R-CNN? As you know, in R-CNN the input image is divided into region proposals and CNN had to process each region, which takes a large computation, instead of this Fast R-CNN sends the input image directly into CNN and CNN will need to process the image only once and generate an output vector in the form of feature map for the whole image at once; this makes the Fast R-CNN faster than R-CNN.

5.1.3 Faster R-CNN:

Although Fast R-CNN is fast it still uses a selection search to detect region proposals. As selection search is a slow algorithm it still makes the Fast R-CNN slow. Ross Girshick et al came up with an idea to improve the Fast R-CNN by replacing the selection search algorithm. He discovered that the feature map generated by CNN can also be used to generate region proposals. So now there is no need to use a selection search algorithm because CNN can do both feature mapping and region selection.

The basic architecture of Faster R-CNN is the same as Fast R-CNN, the input image is fed into CNN which generates a feature map and another CNN is used to predict the region proposals. After this shared network the output of both networks is combined and sent to the final detection layers (i.e. RoI Pooling and fully connected layers). Using this sharing technique, the computation of region proposals is almost cost-free. For dealing with different shapes and sizes of the detection window, this method uses special anchor boxes instead of using a pyramid of scaled images or a pyramid of different filter sizes. The anchor boxes function as reference points to different region proposals centered on the same pixel. Now because there is no selection searching in Faster R-CNN it has become faster than both Fast R-CNN and R-CNN.

For a real-world application, object detection and characterization are significant computer vision tasks in artificial intelligence. You need to distinguish different objects like vehicles, human beings, animals, and other items and their location as well. Earlier a sliding window was used in object detection models to find where the object was encased and made a boundary box around it. Models like CNN and fast R-CNN are tedious and require and require extensive computation which makes it hard to train these models since every part must be trained independently. In 2015 Joeseph Redmon concocted YOLO (You Look Only Once) that showed object detection as a regression problem instead of a classification

5.1.4 YOLO

YOLO has proved itself as an extremely quick technique to recognize articles. It processes images progressively at 45 frames per second to predict boundary boxes and classification probabilities at the same time which makes YOLO pretty basic. YOLO does not use any sliding window; unlike other techniques, it sees the whole image at the time of training and testing from which it encodes relevant data regarding classes. YOLO predicts all boundary boxes at once utilizing these extracted features. The picture is then partitioned to make the SxS grid network, and each brace introduces a confidence score corresponding to its boundary boxes. This score shows the accuracy of its prediction and how sure the model is that the boundary box carries an object in it. Including a confidence score, there are four other predicting parameters: x,y,w, and h. Here x and y are the coordinates that represent the center of the box, and w and h are the predicted width and height of the box Compared to the entire image to extract features of the image initial convolution layers are used, and the output is predicted by fully connected layers. It consists of 24 convolutional layers along with 2 fully connected layers.

Since YOLO v1 was facing some localizing and low recall predicting errors. YOLO v2 was introduced with some improvements. YOLO v2 contains batch normalization in all convolutional layers. The batch normalization gives a regularization effect which improves its performance by 2% mAP. Secondly, it trains the classifier on low resolution and then calibrates it from 224x224 to 448x448 for 10 epochs, as higher resolution inputs are conformed by network filters. YOLOv2 uses anchor boxes which have aMELiorated the recall by 7% which has resultantly increased the level of positive cases. YOLOv2 utilizes k-means clustering to get a good intersection over union (IOU) score. Using the activation function YOLOv2 limits the value of location between 0 and 1 resultantly increasing 5% mAP. For detecting the smaller objects, the original 13x13 feature map is mapped to a 13x13x2048 feature map. Other than these modifications YOLOv2 has darknet-19 as well for feature extraction. To decrease the number of parameters darknet-19 has numerous 1x1 convolutions which help to maintain good harmony between the complexity and accuracy of the model.

It is an upgraded version of YOLOv2, enlivened from ResNet and as it contains skip associations and 3 forecast heads like a feature pyramid network (FPN) it is also called darknet-53 because it has 52 convolutions. It consists of a total of 106 layers, half of them are trained in ImageNet and the other half are attached to it for article detection which makes it better and stronger yet slower at the same time. Instead of utilizing SoftMax which was used in YOLOv2, it uses logistic classifiers for the mean of multiclass labeling. YOLOv3 applies three 1x1 detection kernels in different places and of different sizes for identification purposes. As a result of these modifications localizing errors were reduced, and the accuracy of detecting small objects and notable mAP has increased. But still, yolov3 could not get better results on small objects as compared to Faster R-CNN.

YOLOv1 has a framework of darknet trained Imagenet-1000 and YOLOv2 come with a feature extractor of darknet-19 but the overall accuracy of the model is increased in YOLOv3 which has darknet-53. YOLOv1 was not good at detecting and localizing small objects or object in-group, whereas with the use of anchor boxes YOLOv2 was able to detect small objects and this detection capability was then more modified in YOLOv3 using residual block.

5.2 Proposed Methodology

In this work we have classified the bird's sound using the proposed architecture shown in Fig. 34. To apply the proposed model, we have to modify the data according to the model input. We converted the sound data into the mal Spectrogram and that mal Spectrogram will also be scaled to distinguish between higher frequencies and simpler to do so for lower frequencies. Our model is based on the CNN we need the data in the image form so we convert the data into 2-D Images. The proposed method comprises four different stages:

- In the first step, data is preprocessed by using multiple techniques as we need to convert it for model input. The data is the algorithm that retrieves energies of log-MEL scaled from the sound signals. It normalizes as well as preprocesses them.
- The second step generating the Spectrogram of the voice dataset
- The third step we passed to the proposed model for the classification process. Figure 34. displays the overall structure of the proposed.

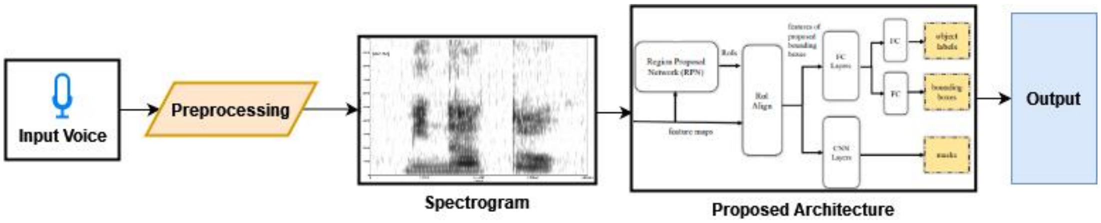

Figure 28. Proposed Methodology for Bird Sound Classification using Mask-R-CNN

5.3 Bird's Sound Data Preprocessing

We have performed extensive Exploratory Data Analysis where we analyze the metadata and sound waves. Now we have to train a model named Mask-RCNN where the model will classify and localize the exact species of the birds during singing. We first perform metadata analysis and now we have to perform to analyze the sound waves of the bird. There are about 264 bird species in the dataset and for each species multiple recordings are present. We will be demonstrating the random bird chirps recorded from the dataset and its sound plot.

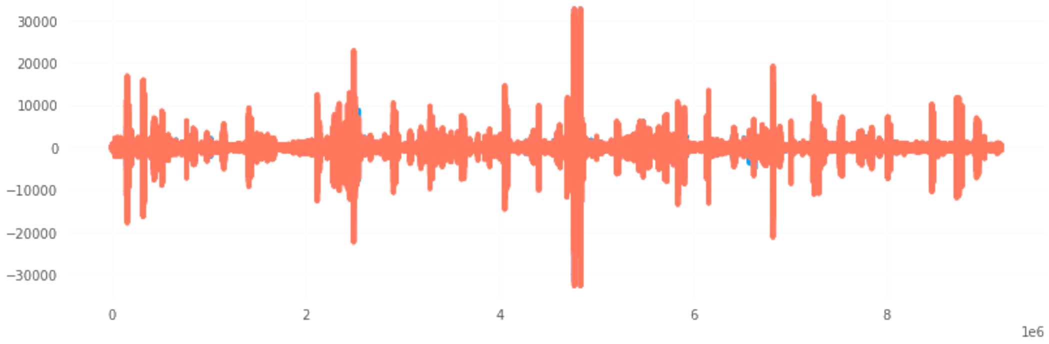

Figure 29. Snow Bunting Bird Wave plot

5.3.1 Extracting Features from Sounds

The following key features are eliminated from the bird’s sound file and then fed into our deep learning model.

5.3.1.1 Eliminating muted starting and Ends

The muted portions are eliminated from the beginning and the end of the sound signal. At one end of several environmental sounds, there is silence that doesn't convey any meaningful information. Since along the time axis, the non-silent part of the audio can be shifted freely once the mute element is removed the augmented signal accepts more variation

5.3.1.2 Random Scaling

A sound wave's linear interpolation is performed using a random scaling factor that is uniformly sampled over the range [1.25-1, 1.25]. Time stretching and pitch shifting collectively is similar to this. The method is only applied since it utilizes the wave and proves to be much quicker and less costly which individually requires two short-time Fourier transformations.

5.3.1.3 Random Padding/Random Crop

The duration of the sound and length of the signal is altered by scaling and eliminating the muted part. Based on the augmented signal length random padding and cropping are applied to create a uniform length of the signal.

The audio data is composed by:

Sound: Sequence of vibrations in varying pressure strengths (y)

Sample Rate: (SR) is the number of samples of audio carried per second, measured in Hz or kHz.

A Python library for music and audio analysis is called Librosa. It offers the building elements required to develop music information retrieval systems. Using several signal processing techniques, Librosa assists with the visualization of audio signals as well as feature extraction from them.

The path to an audio file will be read in by the Librosa load function, which will then return a tuple containing two components. An "audio time series" (type: array) corresponding to an audio track is the first item. The sample rate that was applied to the audio processing is the second element of the tuple. Fig. 31 shows different bird species' preprocessed audio.

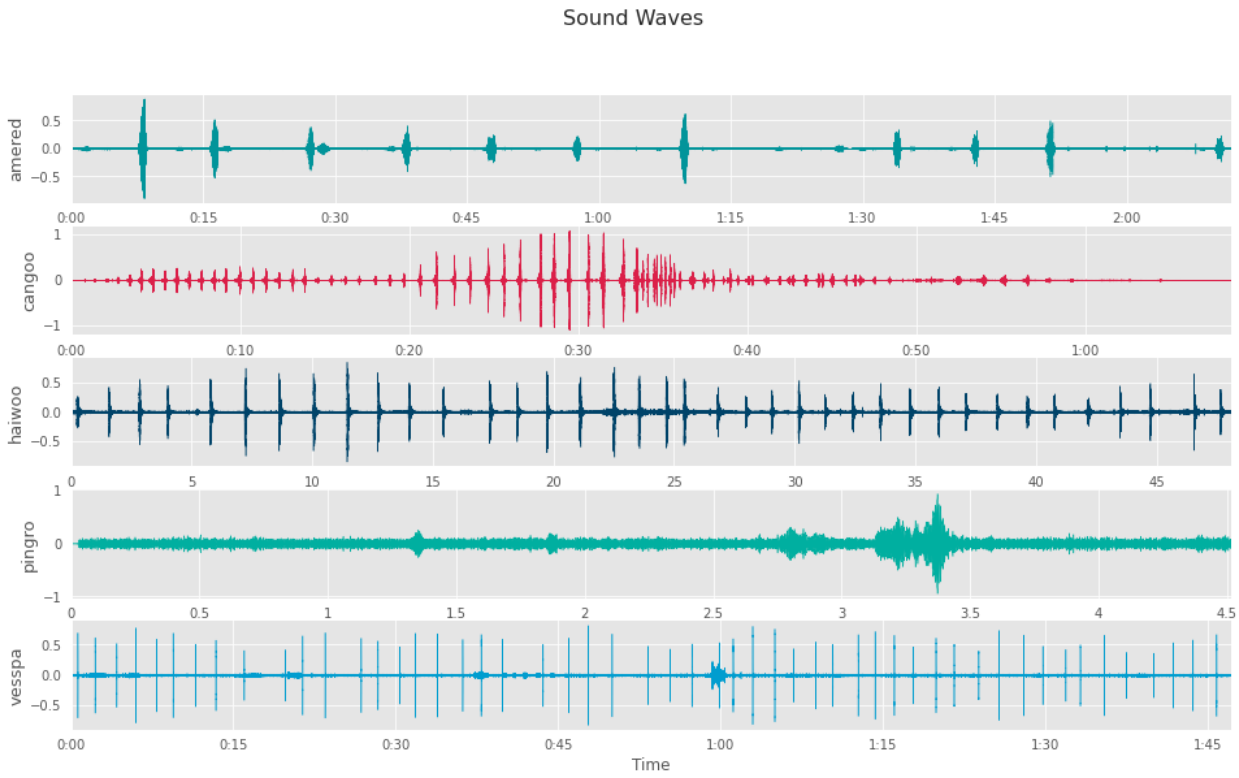

Figure 30. Librosa Sound wave visualization for 5 different Birds

Librosa uses a 22050-sampling rate by default, but you may enter virtually any other sampling rate you want. Be careful: Resampling could increase the load function's execution time by a substantial amount (depending on your task). With the default settings, I discovered that loading Stairway, an 8-minute song, took around 16 seconds.

5.3.2 MEL Spectrogram

A MEL is a numerical value that characterizes a pitch, just like a frequency does. A4 has a frequency of 440 Hz if we take it as an example. At A5, the frequency doubles to 880 Hz, and at A6, it doubles once more to 1760 Hz. Therefore, there is a leap of 440 between the A4 and A5, and 880 between the A5 and A6, but the human ear doesn't hear in that manner.

Librosa has two functions: one to easily show the resultant MEL Spectrogram and another to extract the power Spectrogram (amplitude squared) for each MEL over time. It is necessary to import this display function separately because it is not automatically loaded with Librosa. The sound MEL Spectrogram for the 5 different bird's voices is shown in Fig. 32. We can examine the Spectrograms of the selected 5 bird species, and we can note that as time goes on, the frequencies of each species have various patterns. For example, the vespa has a high frequency, while the cangoo has diverse patterns.

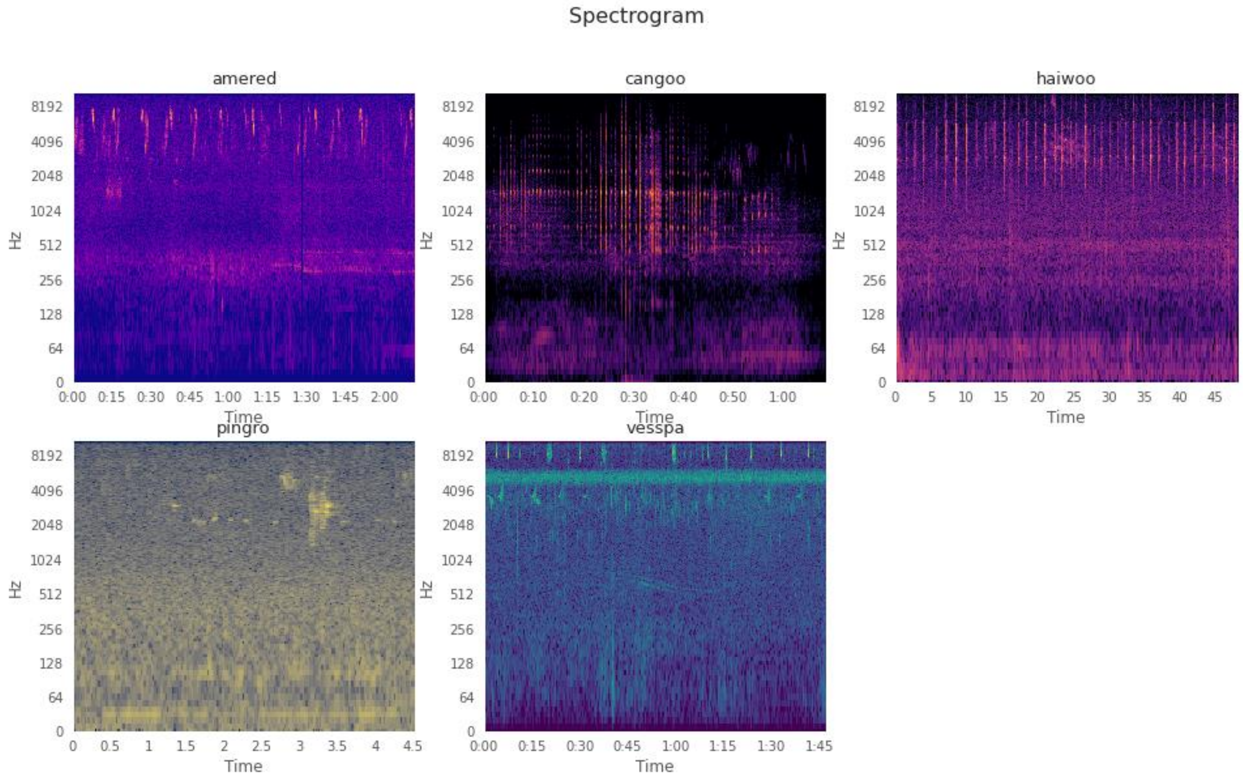

Figure 31. Amplitude Spectrogram for 5 different bird's voices

5.3.3 MEL Scale

The frequency of a signal is transformed logarithmically to create the MEL Scale. The fundamental tenet of this transformation is that human ears perceive sounds with equal distances on the MEL Scale as having equal distances. In reality, individuals find it more difficult to distinguish between higher frequencies and simpler to do so for lower frequencies. Therefore, despite the fact that the distance between the two sets of sounds is the same, we do not perceive it as such. The MEL Scale's ability to accurately simulate our hearing is what makes it a crucial tool for Machine Learning applications to audio. We converted MEL Spectrograms to MEL Scale to find a clear graph for the respective species to classify into their respective classes using deep learning methods. Fig. 33 shows the MEL Scale for 5 different birds' voices.

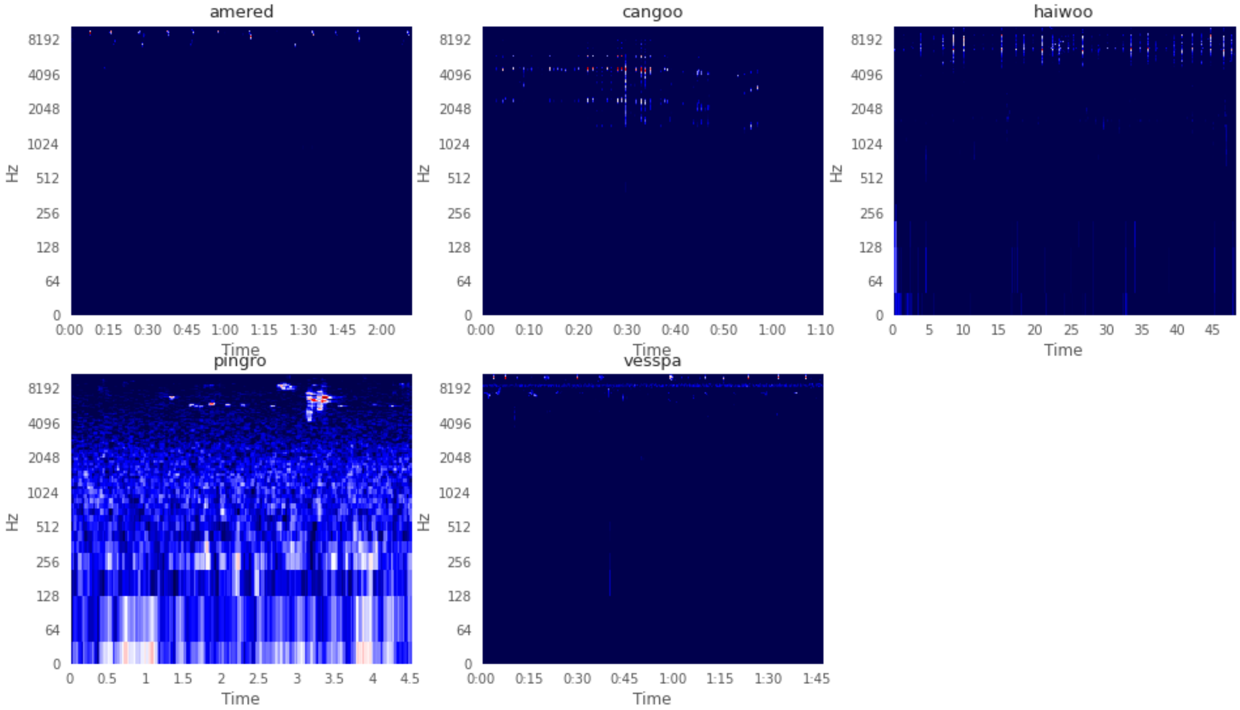

Figure 32. MEL Scaled Spectrogram for 5 different bird's voices

5.3.4 Converting into 2-D Images

The Spectrogram is a brief "snap picture" of an audio wave, and since it is an image, it may be easily supplied to CNN-based structures designed to handle images. To prepare the dataset, we all the first and last silent time stamp from the audio and then converted them into a Spectrogram to perform further analysis and improve the Spectrogram representation. The processed MEL Spectrogram is further converted to a 2-D image of a fixed size for each folder of the wave file from the xento-canto database. This dataset is used for classification purposes using the DesnseNet convolutional neural network.

5.4 Feature Extraction

Log-MEL Spectrograms are preprocessed and normalized before being fed as input to the model. Two-dimensional visualizations of sound in the time-frequency domain as Spectrograms, having luminance or color signifying the intensity of individual frequencies in frame concerning time. Compared to manually extracted features, these preprocessed features carry more information. The feature map generated by log MEL is located in the domain of time as well as frequency. For identifying sound occurrences through deep learning models, MEL-Spectrogram has shown excellent characteristics. They vary from standard Spectrograms as log MEL-scale filter banks are being utilized to simulate the impression of sounds like a human ear. 46ms window is applied to retrieve a MEL filter bank of size 128 for each sound signal which is overlapped by half size of the original signal. The filter bank of the MEL-Spectrogram and carry an equal size to the window. The filter bank values of these models are then evaluated and normalized.

5.5 Region-Proposal with Mask R-CNN

Surrounding Spectrograms, recognizing bounding boxes efficiently is done by Mask-RCNN which may directly relate to the output. In computer vision, being a fully convolutional network Mask-RCNN has proved its excellence in identifying objects as well as segmentation. For each targeted object, it returns a mask, bounding box, and labels for its category. The general working of Mask-RCNN is shown in Figure 3.2. Mask RCNN has two stages for identifying objects. the first stage generated estimated bounding boxes in the image for targeted objects through Region Proposal Network (RPN). Segmentation masks, tight bounding boxes, and labels for the object's category are generated in the second stage. The architecture of Mask RCNN is shown in Figure. 35.

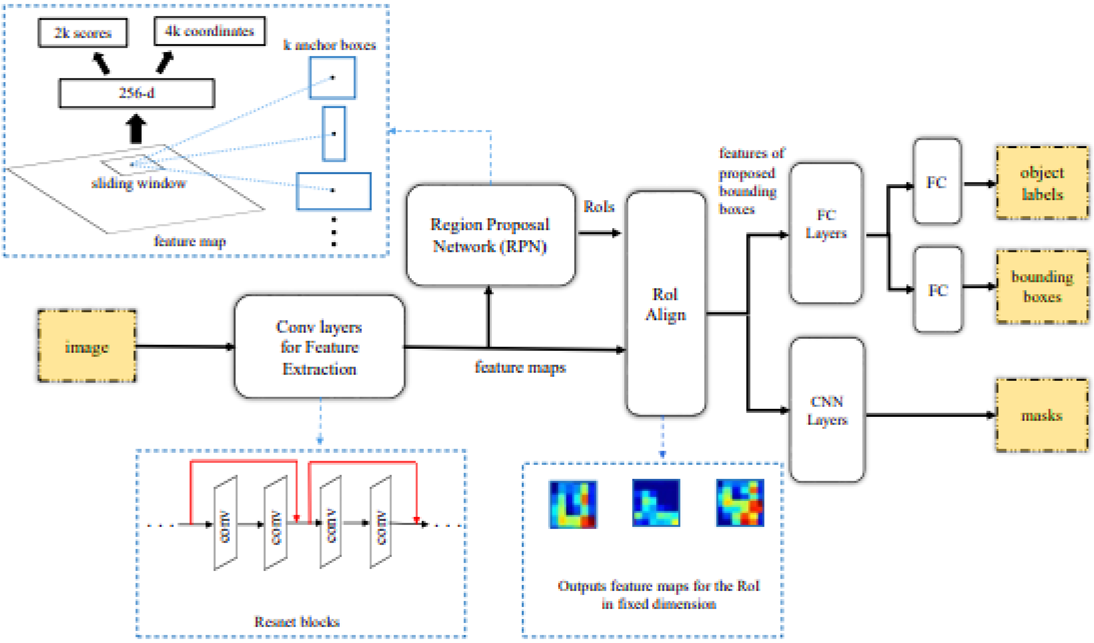

Figure 33. General Mask RCNN Architecture

High-level features are extracted from the images by the backbone architecture of the model which uses ResNet-202 for this purpose. To generate region proposals and scores of object-ness, a small network window is slid above the feature map. For each window that is sliding over, a region proposal is indicated resulting in coordinates by 4k values and probability of presence or absence of object by 2k for an individual proposed regular region

Before getting fed to the convolutional layers, a feature map must define the varying dimensions of the proposed regions with fixed lengths. This fixed length features maps are generated by Mask-RCNN using a Roi Align while making sure that these values are related to the regions of the original file. When RoI isn't aligning along the feature map values, accurate values are then computed by utilizing bi-linear interpolation rather than employing harsh quantization in Faster-RCNN (Ren et al. n.d.). The way Roi Align uses bi-linear interpolation is to improve feature prediction for object boxes.

The ROI Align technique of Mask RCNN ensures that masks, as well as bounding boxes, are aligned with each pixel of an individual object. This results in a significant enhancement in the accuracy of object detection. This ability of Mask R-CNN is one of the main reasons why it is ideal for SED, where identifying the precise commencement of the occurrence is critical. The segmentation branch isn't used in our work since constructing segmentation masks for sounds such as 'gunshot' is difficult as sound occurrences don't have defined bounds and are additionally not supplied in existing data sets.

5.6 Frame-Level Classifier and Post-Processing

Mask R-CNN produces a lot of plausible event regions with precise bounds, but it can produce a lot of false positives as well. it happens because of nonlinear forms in the time-frequency context of the occurrence which might result in mistakenly identifying noise from the background as the target event. Short events like gunshots face this issue severely because their MEL Spectrogram depicts high variation. A classifier is hence trained which examines the sound segments shown by log-MEL values to determine if the event belongs to a certain class or not to keep the actual events from the estimated regions. The log-MEL frequency vectors of several successive time increments made up these segments. These vectors are known as frames. Depending on if a frame is background noise or an actual occurrence 0 and 1 labels are given to it. A one-dimensional vector representing the chance of an individual frame of the segment is the part target event show frame-level classifiers' outcome.

6 Experimental Analysis, Results, and Conclusion

6.1 Experimental Analysis

For experimental analysis, we used Jupyter notebook as python IDE for coding. The hardware we had for performing these experiments includes 64 GB RAM and Nvidia RTX 3080 GPU. We used the Xento-canto bird voices dataset which we already discussed in detail in Chapter 3. Here we discuss its audio features and available files for training and testing purposes of our model. It provides access to audio recordings of wild birds from all around the world through an internet database. Thousands of recorders from across the world, both amateur and expert birdwatchers, exchange their recordings in a developing online community. The goal of Xeno-canto is to depict every bird sound, including every taxon, every subspecies, every aspect of regional variety, and every developmental stage. The dataset contains several files including train audio that consists of the short recording of different birds uploaded by community members. Test Audio is about 150 mp3 recordings, each lasting about 10 minutes, are concealed in the test audio directory. They won't all fit in the memory of a laptop at once. Three different isolated sites in North America are where the recordings were made. Sites 1 and 2 were labeled at 5-second intervals and need matching predictions, while site 3 files are only labeled at the file level due to the time-consuming labeling procedure. As a result, site 3 has a test set with fewer rows and requires predictions with lower temporal resolution. There are also CSV files of test and train, test CSV contains site id, row id, and seconds and audio id. While train CSV has metadata it provides ebird code, recordist, location, date, and file name. Both train and test have metadata example test audio and train audio, test audio has labels with higher time precision than hidden test sets. While example test audio summary has filename seconds, birds encodes, filename and seconds records.

6.2 Model Implementation

In this section, we will discuss the Mask RCNN model implementation and discuss how it will fit for classification and detection tasks. Also, how we trained our model with only weak annotation. Before building the model, we have to identify the bird's sound from a continuous (long) audio clip and estimate when each bird's sound would occur. As a result, this is how we anticipate the model-building task will turn out. When it comes to audio tagging, we must forecast the speech command that will be heard in each audio clip at the clip level (which is in a sense similar to the Audio Tagging task, because we only need to provide clip level prediction).

With knowledge of the onset and offset times, our model can make predictions. Because of this, bird sound classification models output segment-by-segment predictions rather than the clip-by-clip predictions that Audio Tagging models typically provide. Consider a 2D CNN-based model where the input is a log-MEL Spectrogram, features are extracted using the CNN feature extractor, and classification is performed using the feature map produced by CNN. If we aggregate the CNN feature extractor's output simply along the frequency axis, we can still keep time information on that feature map because the output should be four-dimensional (batch size, channels, frequency, and time). The feature map shows which bird sound occurs in which time segment.

Before implementing models on the dataset, we perform preprocessing on data, for that purpose we wrote a custom class. Audio file processing and features extraction of duration and file types (DFT) are by using torch Librosa. DFT class consists of multiple functions including a function for finding the matrix and the inverse of the matrix. Implementation of STFT with Conv1d is the function that has the same output as Librosa. Another class Spectrogram is there that calculates Spectrogram using Pytorch to calculate logMEL Spectrogram. We further Interpolate data in the time domain. This is used to compensate for the resolution reduction in the down sampling of a CNN. The good thing about Mask RCNN models is that they accept raw audio clips as input. After that put chunk into the CNN feature extractor of the model. In the model, the input raw waveform will be converted into log-MEL Spectrogram using torchLibrosa's utilities. We put this functionality in Mask RCNN preprocess method. Mask-RCNN feature extractor method will take this as input and output feature map. Although it's downsized through several convolutions and pooling layers, the size of its third dimension is 15 and it still contains time information. Each element of this dimension is a segment. In the bird sound classification model, we provide a prediction for each of these.

Our scene merely has little annotation (clip-level annotation). For our model, weakly supervised training is consequently necessary. We also need to aggregate the data along the time axis because our weakly-supervised weak annotation only goes up to the clip level. To do this, we first add a classifier that outputs the likelihood that a certain class exists at each time step and then display the classifier's output along the time axis. This enables us to obtain forecasts at the clip and segment levels (if the time resolution is high, it can be treated as an event-level prediction). Then, we train it frequently using BCE loss with clip-level annotation and prediction. The model's AttBlock is where segment- and clip-wise predictions are computed. In the forward method, it at first calculate self-attention map in the first line norm_att = torch.softmax(torch.clamp(self.att(x), -10, 10), dim=-1). This will be used to aggregate the classification result for the segment. In the second line, cla = self.nonlinear_transform(self.cla(x)) calculates segment wise classification result. Then in the third line, attention aggregation is performed to get the clip-wise prediction.

6.3 Evaluation Metric

For evaluating our model performance, we used mAP, loss, and f1 scores as the evaluation metric. Mean Average Precision (mAP) is used to assess detection algorithms like Fast R-CNN, YOLO, Mask R-CNN, etc. Recall values between 0 and 1 are used to determine the average precision (AP) values. We also used a custom loss function and a callback function. We used Stratified Fold cross-validation that returns stratified folds. The folds are made by preserving the percentage of samples for each class. Stratified Fold is used when a need is to balance of percentage each class in train & test. The number of train folds is 17081 and the validation fold is 4271. We set the learning rate to 0.0010 and momentum to 0.9000 and the results are shown in Table 9.

Table 9: Evaluation Parameters on Different Epochs in Training

Epochs	Train/Val	epoch_f1	epoch_mAP	f1	loss	mAP
1	Train	0.0142	0.0066	0.0114	0.0385	0.0304
	Val	0.0419	0.0704	0.0141	0.0249	0.0105
2	Train	0.1051	0.0865	0.0793	0.0204	0.0871
	Val	0.2098	0.2634	0.0454	0.0199	0.0145
3	Train	0.2301	0.2052	0.1616	0.0170	0.1221
	Val	0.3238	0.3840	0.0659	0.0177	0.0156
4	Train	0.3194	0.3032	0.2271	0.0149	0.1402
	Val	0.3785	0.4385	0.0771	0.0161	0.0161
5	Train	0.3858	0.3760	0.2763	0.0135	0.1498
	Val	0.4295	0.4869	0.0882	0.0151	0.0162
6	Train	0.4356	0.4362	0.3155	0.0125	0.1583

	Val	0.4740	0.5162	0.0988	0.0146	0.0165
7	Train	0.4871	0.4908	0.3563	0.0116	0.1642
	Val	0.4951	0.5396	0.1041	0.0139	0.0167
8	Train	0.5230	0.5307	0.3856	0.0110	0.1667
	Val	0.5086	0.5425	0.1084	0.0136	0.0166
9	Train	0.5406	0.5515	0.3994	0.0106	0.1695
	Val	0.5257	0.5604	0.1124	0.0134	0.0166
10	Train	0.5446	0.5664	0.4045	0.0105	0.1702
	Val	0.5221	0.5618	0.1135	0.0133	0.0167

After running training and saving the best checkpoint as weight. We try to get a prediction on different audio clips by passing it to the model weight. Some of the sample predictions are discussed in bellow table 10.

Table 10: Sample Prediction on Trained Model

ebird code	**Onset**	**Offset**	**Max confidence**	**Mean confidence**
aldfly	0.96	2.23	0.985395	0.897037
aldfly	7.04	7.67	0.526610	0.519470
aldfly	11.20	12.15	0.956318	0.928820
aldfly	14.40	15.03	0.809643	0.805571
aldfly	20.16	21.43	0.987058	0.917030
aldfly	18.24	19.19	0.885321	0.789641
aldfly	19.84	22.07	0.983291	0.913019
aldfly	23.04	23.67	0.669237	0.624711

aldfly	25.92	26.87	0.950051	0.897716
Aldfly	27.84	28.79	0.991087	0.899111

Finally, we performed cleaning on the predicted data frame for better understanding and write predicted results into CSV.

6.4 Comparison with Existing Techniques

The contrast between our proposed MASK-RCNN system and earlier AudioSet tagging techniques is shown in Table 11. Because MASK-RCNN is a conventional CNN with a straightforward architecture and can be compared to earlier CNN systems (Choi, Fazekas, and Sandler 2016) (Kong, Cao, et al. n.d.), we utilize it as the foundational model to examine alternative hyper-parameter combinations for AudioSet tagging. Baseline results for random guesses are as follows: mAP = 0.005. The Google (Gemmeke et al. n.d.) result obtained an mAP of 0.314 when trained with embedding features (Hershey et al. 2017). A feature-level attention neural network improved on the single-level attention and multi-level attention systems' respective mAPs of 0.337 and 0.360 to reach an mAP of 0.369. The mAP for Wang et al (Wang et al. n.d.)investigation of five distinct attention functions was 0.362. The embedding functionalities made available with AudioSet (Gemmeke et al. n.d.) provided the foundation for the solutions. The most current DeepRes system (Ford et al. n.d.) had an mAP of 0.392 and was constructed from waveforms taken from YouTube. The bottom rows of Table IV demonstrate that our suggested MASK-RCNN system outperforms the best of the earlier systems, achieving an mAP of 0.5664. For a fair comparison with the MASK-RCNN system, we create Wavegram-LogMEL-CNN using MASK-RCNN as the foundation. The MASK-RCNN system outperformed the Wavegram-LogMEL-CNN system. The full results are displayed in Table 9 in the following section.

Table 11: Comparison with Existing Methods based on mAP

Methods	**mAP**
Random guess	0.005
Google CNN (Gemmeke et al. n.d.)	0.314
Single-level attention (Kong, Xu, et al. n.d.)	0.337
Multi-level attention (Yu et al. 2018)	0.360

Large feature-level attention (Kong, Yu, et al. n.d.)	0.369
TAL Net (Wang et al. n.d.)	0.362
DeepRes (Ford et al. n.d.)	0.392
Our Proposed Mask R-CNN	0.5664

6.5 Conclusion

In comparison to previous state-of-the-art AudioSet tagging systems, several of our suggested Mask-RCNN surpass them. MASK-RCNN achieves an mAP of 0.5664, while ResNet38 earns an mAP of 0.434, outperforming Google's baseline of 0.314. MobileNets are compact systems with fewer multi-adds and parameter numbers. MobileNets are compact systems with fewer multi-adds and parameter numbers. The mAP for MobileNetV1 is 0.389. One-dimensional CNNs such as DaiNet (Dai et al. n.d.)and LeeNet11 (Lee et al. 2019)perform worse than our modified system Res1dNet31, which achieves an mAP of 0.365. Mask-RCNN is a pre-trained model that may be applied to new auditory pattern recognition tasks. Mask-RCNN was used for six audio pattern identification tasks after being trained on the AudioSet dataset. We demonstrate that tuned Mask-RCNN outperforms conventional Mask-RCNN in the ESC-50, MSOS, and RAVDESS classification tasks and come close in the DCASE 2018 Task 2 and GTZAN classification tasks. The trials demonstrate that, even with a small amount of training data, Mask-RCNN is capable of generalizing to additional auditory pattern recognition tasks. We have demonstrated pre-trained audio neural networks (Mask-RCNN) for audio pattern detection that were trained on the AudioSet. To create Mask-RCNN, a variety of neural networks are studied. We provide a model for audio set tagging that achieves cutting-edge performance and archives an mAP of 0.5664. We also look into Mask-RCNN's computational complexity. We demonstrate that Mask-RCNN outperforms multiple earlier state-of-the-art systems in a variety of audio pattern identification tasks. When Mask-RCNN are fine-tuned on a little quantity of data for brand-new tasks, they can be helpful. We will expand Mask-RCNN to more auditory pattern recognition jobs in the future.

7 References

Aggarwal, Rajeev, Jai Karan Singh, Vijay Kumar Gupta, Sanjay Rathore, Mukesh Tiwari, and Anubhuti Khare. 2011. “Noise Reduction of Speech Signal Using Wavelet Transform with Modified Universal Threshold.” *International Journal of Computer Applications* 20(5):14–19. doi: 10.5120/2431-3269.

Anon. n.d.-a. “Deep Learning. Book in Preparation for MIT Press - Google Scholar.”

Anon. n.d.-b. “TensorFlow for Deep Learning: From Linear Regression... - Google Scholar.”

Atrey, PK, … NC Maddage-2006 IEEE International, and undefined 2006. n.d. “Audio Based Event Detection for Multimedia Surveillance.” *Ieeexplore.Ieee.Org*.

El Ayadi, Moataz, Mohamed S. KaMEL, and Fakhri Karray. 2011. “Survey on Speech Emotion Recognition: Features, Classification Schemes, and Databases.” *Pattern Recognition* 44(3):572–87. doi: 10.1016/J.PATCOG.2010.09.020.

Barchiesi, Daniele, Dimitrios Giannoulis, Dan Stowell, and Mark D. Plumbley. 2014. “Acoustic Scene Classification: Classifying Environments from the Sounds They Produce.” *Ieeexplore.Ieee.Org*.

Boureau, YL, J. Ponce, … Y. LeCun-of the 27th international conference on, and undefined 2010. n.d. “A Theoretical Analysis of Feature Pooling in Visual Recognition.” *Di.Ens.Fr*.

Brandes, T. Scott. n.d. “Automated Sound Recording and Analysis Techniques for Bird Surveys and Conservation.” doi: 10.1017/S0959270908000415.

Breiman, Leo. 2001. “Random Forests.” 45:5–32.

Cakir, Emre, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. 2015. “Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks.” *Proceedings of the International Joint Conference on Neural Networks* 2015-Septe. doi: 10.1109/IJCNN.2015.7280624.

Caliskan, A., H. Badem, … A. Basturk-IU-Journal of Electrical &., and undefined 2017. 2017. “Diagnosis of the Parkinson Disease by Using Deep Neural Network Classifier.” *Dergipark.Org.Tr* 17(2):3311–18.

Casale, S., A. Russo, G. Scebba, and S. Serrano. 2008. “Speech Emotion Classification Using Machine Learning Algorithms.” *Proceedings - IEEE International Conference on Semantic Computing 2008, ICSC 2008* 158–65. doi: 10.1109/ICSC.2008.43.

Chandrakala, S., and S. L. Jayalakshmi. 2020. “Environmental Audio Scene and Sound Event Recognition for Autonomous Surveillance.” *ACM Computing Surveys* 52(3):1–34. doi: 10.1145/3322240.

Choi, Keunwoo, György Fazekas, and Mark Sandler. 2016. “Automatic Tagging Using Deep Convolutional Neural Networks.” *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016* 805–11.

Choi, Tacksung, Sunkuk Moon, Young Cheol Park, Dae Hee Youn, and Seokpil Lee. 1996. “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models.” *IEEE Trans. Speech Audio Process.* 3(1):47–60. doi: 10.1587/TRANSINF.E92.D.1584.

Chu, Selina, Shrikanth Narayanan, and C. C. Ja. Kuo. 2009. “Environmental Sound Recognition with TimeFrequency Audio Features.” *IEEE Transactions on Audio, Speech and Language Processing* 17(6):1142–58. doi: 10.1109/TASL.2009.2017438.

Dai, W., C. Dai, S. Qu, J. Li, S. Das-2017 IEEE international, and undefined 2017. n.d. “Very Deep Convolutional Neural Networks for Raw Waveforms.” *Ieeexplore.Ieee.Org*.

Dang, A., TH Vu, JC Wang-2017 International Conference on, and undefined 2017. n.d. “A Survey of Deep Learning for Polyphonic Sound Event Detection.” *Ieeexplore.Ieee.Org*.

Dekkers, Gert, Steven Lauwereins, Bart Thoen, Weldegebreal Adhana, Henk Brouckxon, Bertold Van Den Bergh, Toon Van Waterschoot, Bart Vanrumste, Marian Verhelst, Peter Karsmakers, and Mulu Weldegebreal Adhana. 2017. “The SINS Database for Detection of Daily Activities in a Home Environment Using an Acoustic Sensor Network.” *Lirias.Kuleuven.Be*.

Dobre, R. A., V. A. Nita, A. Ciobanu, C. Negrescu, and D. Stanomir. 2015. “Low Computational Method for Siren Detection.” *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging, SIITME 2015* 291–95. doi: 10.1109/SIITME.2015.7342342.

Drossos, K., S. Lipping, T. Virtanen-ICASSP 2020-2020 IEEE, and undefined 2020. 2019. “Clotho: An Audio Captioning Dataset.” *Ieeexplore.Ieee.Org*.

Ford, L., H. Tang, F. Grondin, JR Glass- InterSpeech, and undefined 2019. n.d. “A Deep Residual Network for Large-Scale Acoustic Scene Analysis.” *Groups.Csail.Mit.Edu*.

Gemmeke, Jort F., Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. n.d. “Audio Set: An Ontology and Human-Labeled Dataset for Audio

Events." *Ieeexplore.Ieee.Org*.

GnaMELe, N'tcho Assoukpou Jean, Yelakan Berenger Ouattara, Toka Arsene Kobea, Geneviève Baudoin, and Jean Marc Laheurte. 2019. "KNN and SVM Classification for Chainsaw Sound Identification in the Forest Areas." *International Journal of Advanced Computer Science and Applications* 10(12):531–36. doi: 10.14569/IJACSA.2019.0101270.

Graves, A., A. Mohamed, G. Hinton-2013 IEEE international, and undefined 2013. n.d. "Speech Recognition with Deep Recurrent Neural Networks." *Ieeexplore.Ieee.Org*.

Heittola, Toni, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. 2013. "Context-Dependent Sound Event Detection." *Eurasip Journal on Audio, Speech, and Music Processing* 2013(1). doi: 10.1186/1687-4722-2013-1.

Hershey, Shawn, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. 2017. "CNN Architectures for Large-Scale Audio Classification." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 131–35. doi: 10.1109/ICASSP.2017.7952132.

Hinton, GE, S. Osindero, YW Teh-Neural computation, and undefined 2006. n.d. "A Fast Learning Algorithm for Deep Belief Nets." *Direct.Mit.Edu*.

Hoshen, Y., RJ Weiss, KW Wilson-2015 IEEE international, and undefined 2015. n.d. "Speech Acoustic Modeling from Raw Multichannel Waveforms." *Ieeexplore.Ieee.Org*.

Ilievski, I., T. Akhtar, J. Feng, C. Shoemaker-Proceedings of the AAAI, and undefined 2017. n.d. "Efficient Hyperparameter Optimization for Deep Learning Algorithms Using Deterministic Rbf Surrogates." *Ojs.Aaai.Org*.

Jaitly, N., G. Hinton-2011 IEEE International Conference on, and undefined 2011. n.d. "Learning a Better Representation of Speech Soundwaves Using Restricted Boltzmann Machines." *Ieeexplore.Ieee.Org*.

Juang, B. H., and Lawrence R. Rabiner. 2004. "Automatic Speech Recognition-A Brief History of the Technology Development."

Kahl, Stefan, Thomas Wilhelm-Stein, Hussein Hussein, H. Klinck, D. Kowerko, M. Ritter, and Maximilian Eibl. 2017. "Large-Scale Bird Sound Classification Using Convolutional Neural Networks." *Undefined*.

Kong, Q., Y. Cao, T. Iqbal, Y. Xu, … W. Wang-arXiv preprint arXiv, and undefined 2019. n.d. “Cross-Task Learning for Audio Tagging, Sound Event Detection and Spatial Localization: DCASE 2019 Baseline Systems.” *Arxiv.Org*.

Kong, Q., Y. Xu, … W. Wang-2018 IEEE International, and undefined 2018. n.d. “Audio Set Classification with Attention Model: A Probabilistic Perspective.” *Ieeexplore.Ieee.Org*.

Kong, Q., C. Yu, Y. Xu, T. Iqbal, … W. Wang-…. /ACM Transactions on, and undefined 2019. n.d. “Weakly Labelled Audioset Tagging with Attention Neural Networks.” *Ieeexplore.Ieee.Org*.

Krizhevsky, A., I. Sutskever, GE Hinton-Communications of the ACM, and undefined 2017. 2017. “Imagenet Classification with Deep Convolutional Neural Networks.” *Dl.Acm.Org* 60(6):84–90. doi: 10.1145/3065386.

Kumar, Anurag, and Bhiksha Raj. 2016. “Audio Event Detection Using Weakly Labeled Data.” *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference* 1038–47. doi: 10.1145/2964284.2964310.

Lee, Jongpil, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. 2019. “Sample-Level Deep Convolutional Neural Networks for Music Auto-Tagging Using Raw Waveforms.” *Proceedings of the 14th Sound and Music Computing Conference 2017, SMC 2017* 220–26.

López-de-Ipiña, KarMELe, Jesus Bernardino Alonso, Carlos Manuel Travieso, Jordi Solé-Casals, Harkaitz Egiraun, Marcos Faundez-Zanuy, Aitzol Ezeiza, Nora Barroso, Miriam Ecay-Torres, Pablo Martinez-Lage, and Unai Martinez De Lizardui. 2013. “On the Selection of Non-Invasive Methods Based on Speech Analysis Oriented to Automatic Alzheimer Disease Diagnosis.” *Sensors 2013, Vol. 13, Pages 6730-6745* 13(5):6730–45. doi: 10.3390/S130506730.

McCulloch, Warren S., and Walter Pitts. 1943. “A Logical Calculus of the Ideas Immanent in Nervous Activity.” *The Bulletin of Mathematical Biophysics* 5(4):115–33. doi: 10.1007/BF02478259.

Mesaros, A., T. Heittola, T. Virtanen-2016 24th European Signal, and undefined 2016. n.d. “TUT Database for Acoustic Scene Classification and Sound Event Detection.” *Ieeexplore.Ieee.Org*.

Moon, Todd K. 1996. “The Expectation-Maximization Algorithm.” *IEEE Signal Processing Magazine* 13(6):47–60. doi: 10.1109/79.543975.

Palaz, Dimitri, Ronan Collobert, and Mathew Magimai-Doss. 2013. “Estimating Phoneme Class Conditional Probabilities from Raw Speech Signal Using Convolutional Neural Networks.” *Proceedings of the*

Annual Conference of the International Speech Communication Association, INTERSPEECH 1766–70. doi: 10.21437/interspeech.2013-438.

Peng, YT, CY Lin, … MT Sun-2009 IEEE International, and undefined 2009. n.d. “Healthcare Audio Event Classification Using Hidden Markov Models and Hierarchical Hidden Markov Models.” *Ieeexplore.Ieee.Org*.

Phan, Huy, Student Member, Marco Maaß, Radoslaw Mazur, Alfred Mertins, and Senior Member. n.d. “Random Regression Forests for Acoustic Event Detection and Classification.” *Ieeexplore.Ieee.Org* (1):20–31. doi: 10.1109/TASLP.2014.2367814.

Plinge, A., R. Grzeszick, GA Fink-2014 IEEE International, and undefined 2014. n.d. “A Bag-of-Features Approach to Acoustic Event Detection.” *Ieeexplore.Ieee.Org*.

Purwins, Hendrik, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo Yiin Chang, and Tara Sainath. 2019. “Deep Learning for Audio Signal Processing.” *IEEE Journal on Selected Topics in Signal Processing* 13(2):206–19. doi: 10.1109/JSTSP.2019.2908700.

Radhakrishnan, R., … A. Divakaran-IEEE Workshop on, and undefined 2005. n.d. “Audio Analysis for Surveillance Applications.” *Ieeexplore.Ieee.Org*.

Ren, S., K. He, R. Girshick, J. Sun-Advances in neural, and undefined 2015. n.d. “Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks.” *Proceedings.Neurips.Cc*.

Rosenblatt, F. 1958. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” *Psychological Review* 65(6):386–408. doi: 10.1037/H0042519.

RuMELhart, DE, GE Hinton, RJ Williams- nature, and undefined 1986. n.d. “Learning Representations by Back-Propagating Errors.” *Nature.Com*.

Schölkopf, Bernhard. 1998. “SVMs - A Practical Consequence of Learning Theory.” *IEEE Intelligent Systems and Their Applications* 13(4):18–21. doi: 10.1109/5254.708428.

Schuster-Böckler, Benjamin, and Alex Bateman. 2007. “An Introduction to Hidden Markov Models.” *Current Protocols in Bioinformatics* 18(1):A.3A.1-A.3A.9. doi: 10.1002/0471250953.BIA03AS18.

Schuster, Mike, and Kuldip K. Paliwal. 1997. “Bidirectional Recurrent Neural Networks.” *IEEE Transactions on Signal Processing* 45(11):2673–81. doi: 10.1109/78.650093.

Sejdić, E., I. Djurović, J. Jiang-Digital signal processing, and undefined 2009. n.d. “Time–Frequency Feature Representation Using Energy Concentration: An Overview of Recent Advances.” *Elsevier*.

Sharda, M., and N. C. Singh. 2012. “Auditory Perception of Natural Sound Categories--an FMRI Study.” *Neuroscience* 214:49–58. doi: 10.1016/J.NEUROSCIENCE.2012.03.053.

Shrawankar, Urmila, and Vilas Thakare. 2010. “Noise Estimation and Noise Removal Techniques for Speech Recognition in Adverse Environment.” *IFIP Advances in Information and Communication Technology* 340 AICT:336–42. doi: 10.1007/978-3-642-16327-2_40.

Simonyan, Karen, and Andrew Zisserman. 2015. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Stevens, SS, … J. Volkmann-The journal of the, and undefined 1937. 1937. “A Scale for the Measurement of the Psychological Magnitude Pitch.” *Asa.Scitation.Org* 8(3):185. doi: 10.1121/1.1915893.

Temko, A., C. Nadeu-Pattern Recognition Letters, and undefined 2009. n.d. “Acoustic Event Detection in Meeting-Room Environments.” *Elsevier*.

Tom´, Tomáš, and Tomáš Mikolov. 2010. “Overview Introduction Model Description ASR Results Extensions MT Results Comparison Main Outcomes Future Work Recurrent Neural Network Based Language Model.”

Tomasino, Barbara, Cinzia Canderan, Dario Marin, Marta Maieron, Michele Gremese, Serena D’agostini, Franco Fabbro, and Miran Skrap. 2015. “Identifying Environmental Sounds: A Multimodal Mapping Study.” *Frontiers in Human Neuroscience* 9(OCTOBER):567. doi: 10.3389/FNHUM.2015.00567/BIBTEX.

Valero, Xavier, and Francesc Alias. 2012. “Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification.” *IEEE Transactions on Multimedia* 14(6):1684–89. doi: 10.1109/TMM.2012.2199972.

Vellinga, WP, R. Planqué-CLEF (Working Notes), and undefined 2015. n.d. “The Xeno-Canto Collection and Its Relation to Sound Recognition and Classification.” *Academia.Edu*.

Vincent, E., N. Bertin, … R. Gribonval-IEEE Signal Processing, and undefined 2014. n.d. “From Blind to Guided Audio Source Separation: How Models and Side Information Can Improve the Separation of

Sound.” *Ieeexplore.Ieee.Org*.

Wang, Y., J. Li, F. Metze-ICASSP 2019-2019 IEEE International, and undefined 2019. n.d. “A Comparison of Five Multiple Instance Learning Pooling Functions for Sound Event Detection with Weak Labeling.” *Ieeexplore.Ieee.Org*.

Xia, Xianjun, Roberto Togneri, Ferdous Sohel, Yuanjun Zhao, and Defeng Huang. 2019. “A Survey: Neural Network-Based Deep Learning for Acoustic Event Detection.” *Circuits, Systems, and Signal Processing* 38(8):3433–53. doi: 10.1007/S00034-019-01094-1.

Yu, Changsong, Karim Said Barsim, Qiuqiang Kong, and Bin Yang. 2018. “Multi-Level Attention Model for Weakly Supervised Audio Classification.”

Yu, D., H. Wang, P. Chen, Z. Wei-International conference on rough sets and, and undefined 2014. 2016. “Mixed Pooling for Convolutional Neural Networks.” *Springer* 8818:364–75. doi: 10.1007/978-3-319-11740-9_34.

Yu, Dong, and Li Deng. 2015. “Automatic Speech Recognition.” doi: 10.1007/978-1-4471-5779-3.

Zhang, Xin, Aibin Chen, Guoxiong Zhou, Zhiqiang Zhang, Xibei Huang, and Xiaohu Qiang. 2019. “Spectrogram-Frame Linear Network and Continuous Frame Sequence for Bird Sound Classification.” *Ecological Informatics* 54:101009. doi: 10.1016/J.ECOINF.2019.101009.

Zhang, Y., D. LV-The Open Automation and Control Systems, and undefined 2015. 2015. “Selected Features for Classifying Environmental Audio Data with Random Forest.” *Benthamopen.Com* 7:135–42.

Zhuang, Xiaodan, Xi Zhou, Mark A. Hasegawa-Johnson, and Thomas S. Huang. 2010. “Real-World Acoustic Event Detection.” *Pattern Recognition Letters* 31(12):1543–51. doi: 10.1016/J.PATREC.2010.02.005.