KANTIPUR ENGINEERING COLLEGE

(Affiliated to Tribhuvan University)

Dhapakhel, Lalitpur

[Subject Code: CT707]

A MAJOR PROJECT MID-TERM REPORT ON

FEATHERFIND : BIRD SPECIES IDENTIFICATION FROM AUDIO

Submitted by:

Gaurav Giri	**[Kan077bct034]**
Iza K.C.	**[Kan077bct039]**
Prajwal Khatiwada	**[Kan077bct056]**
Samrat Kumar Adhikari	**[Kan077bct074]**

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

December, 2024

FEATHERFIND : BIRD SPECIES IDENTIFICATION FROM AUDIO

Submitted by:

Gaurav Giri	**[Kan077bct034]**
Iza K.C.	**[Kan077bct039]**
Prajwal Khatiwada	**[Kan077bct056]**
Samrat Kumar Adhikari	**[Kan077bct074]**

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

Kantipur Engineering College

Dhapakhel, Lalitpur

December, 2024

ABSTRACT

This report presents ”FeatherFind”, a comprehensive model designed for identifying bird species using audio recordings. The audio identification process involves collecting and preprocessing bird sound datasets, isolating key features using methods like Mel-Frequency Cepstral Coefficients (MFCC), and employing a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) to accurately recognize bird calls.The approach aims to enhance the accuracy and efficiency of bird species identification, facilitating better conservation efforts, biodiversity monitoring, ecological research, and environmental education. By combining advanced machine learning techniques with extensive datasets, FeatherFind promises to offer a robust solution for the automated identification and monitoring of bird species, particularly those that are endangered, thereby contributing significantly to the field of avian research and conservation.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
GA	Genetic Algorithm
GPS	Global Positioning System
GRU	Gated Recurrent Network
LSTM	Long Short-Term Memory
MAP	Mean Average Precision
MFCCs	Mel-Frequency Cepstral Coefficients
MLP	Multilayer Perceptron
RGB	Reg Green Blue
ROC	Receiver Operating characteristics
RNN	Recurrent Neural Network
STFT	Short-Time Fourier Transform

CHAPTER 1
INTRODUCTION

Globally, the avian kingdom is vast, with over 11,000 species, showing nature's complexity and evolutionary skill. This number, from the International Ornithological Committee as of April 2023, shows the great bird diversity, with each species having its own role and story.[1]

In Nepal, a country known for its rich nature and varied ecosystems, from the lowland Terai to the high Himalayas, there are more than 887 bird species, according to the Himalayan Nature organization. This is over 8% of the world's known bird species, a big number given Nepal's small size.[2]

Many of these species are endangered due to habitat loss, climate change, and human activities. The National Red List of Nepal's Birds identifies 168 nationally threatened bird species, including 68 Critically Endangered, 38 Endangered, and 62 Vulnerable species, as detailed in a publication by the Journal of Threatened Taxa.[3]

The situation of these endangered species shows the need for conservation efforts. Technologies such as audio recognition provide new methods for identifying and monitoring bird populations. By analyzing bird sounds and pictures, researchers can better understand species distribution, behavior, and threats. These technologies not only help conserve endangered species but also support broader biodiversity research.

1.1 Problem Statement

In Nepal, a hotspot of avian biodiversity, accurately identifying and classifying bird species, particularly those that are endangered, is a critical yet complex task. Traditional observation methods are limited by the vast geographical and ecological diversity of the region, making it challenging to monitor and protect these birds effectively. The necessity for precise identification is paramount for conservation efforts aimed at maintaining ecosystem balance. To address this, there is a pressing need for a method that can overcome these constraints by leveraging advanced technologies capable of distin-

guishing between the myriad of bird calls and songs, as well as visual markers through image classification. Such a method promises to automate the identification process, enhancing accuracy and efficiency in monitoring endangered species.

1.2 Objectives

i To develop and implement an integrated technological solution that utilizes advanced audio recognition technique for the accurate identification and monitoring of bird species in Nepal, with an emphasis on endangered species.

1.3 Application Scope

1. **Conservation Efforts:**
   This system will enhance conservation by enabling accurate monitoring of bird populations, helping track endangered species and take a step towards habitat protection.
2. **Biodiversity Monitoring:**
   Automated identification will aid biodiversity monitoring by processing large datasets, helping detect species distribution on bird communities.
3. **Ecological Research:**
   Researchers can use the system to study bird migration, and habitat use, providing crucial data for modeling ecosystems and understanding ecological interactions.
4. **Environmental Education and Awareness:**
   Integrated into educational programs, this tool will raise public awareness about biodiversity and conservation, engaging students and scientists in bird identification.
5. **Bird viewing:**
   Bird enthusiasts will benefit from this system as it will enhance bird watching experiences by providing instant identification of bird species

1.4 Features

1. **Species Identification Using Audio:**

   The app allows users to record bird sounds in real-time using their device's microphone or upload pre-recorded audio files. Advanced noise filtering techniques isolate bird calls from background noise, and sound wave analysis helps in identifying distinct frequency patterns. Machine learning algorithms, trained on a vast database of bird calls, match the recorded sound to identify the bird species accurately.

2. **Mapping Identified Bird Habitat:**

   The app tags the location of identified birds using GPS, providing detailed habitat information typical of each species. Integrating with mapping services, it displays bird sightings on an interactive map, generating heat maps to show species density and distribution. Additionally, it tracks and visualizes bird migration patterns over time, helping users understand seasonal movements.

3. **Provide Description About the Birds:**

   For each identified bird species, the app offers detailed profiles that include scientific and common names, physical descriptions, and conservation status. It also provides audio and visual media for reference, along with information on the bird's behavior, diet, and typical habitats, enriching the user's understanding of the species.

1.5 Feasibility Study

Before implementation of project design, the feasibility analysis of the project must be done to move any further. The feasibility analysis of the project gives an idea on how the project will perform and its impact in the real world scenario. So, it is of utmost importance.

1.5.1 Economic Feasibility

Our system is economically achievable as a result of the development of several tools, libraries, and frameworks. Since all the software required to construct it is free and readily available online, this project is incredibly cost-effective. Only time and effort are needed to create a worthwhile, genuinely passive system. The project doesn't come at a substantial cost. From an economic standpoint, the project appears successful in this sense.

1.5.2 Technical Feasibility

The software needed to implement a project can be downloaded from a wide variety of online resources. Technically speaking, the project is feasible as the necessary software is easily available. We were able to learn the information we required for the project through a variety of online sources, including classes. All the libraries and data are accessible online for free because this project does not require any licensing costs. It is technically possible if one has the necessary information and resources.

1.5.3 Operational Feasibility

The project aims to enhance bird species identification through audio classification, making bird sound recognition more accessible and efficient. This solution is particularly beneficial for ornithologists, bird watchers, and environmental researchers who require accurate and quick identification of bird species based on their calls. The project leverages advanced audio signal processing and deep learning algorithms to classify bird sounds, ensuring high accuracy and reliability. Given the widespread availability of mobile devices and recording equipment, the project is operationally feasible, as it can be easily integrated into existing workflows and tools used by bird enthusiasts and professionals. By providing an efficient method for bird sound classification, the project supports a sizable community interested in avian studies and conservation, ensuring practical applicability and ease of use.

1.5.4 Schedule Feasibility

The workload of the project is divided amongst the project members. The scheduling is done according to an incremental model where different modules are planned to be assigned to the group members.So, the project fulfills the schedule feasibility requirements.

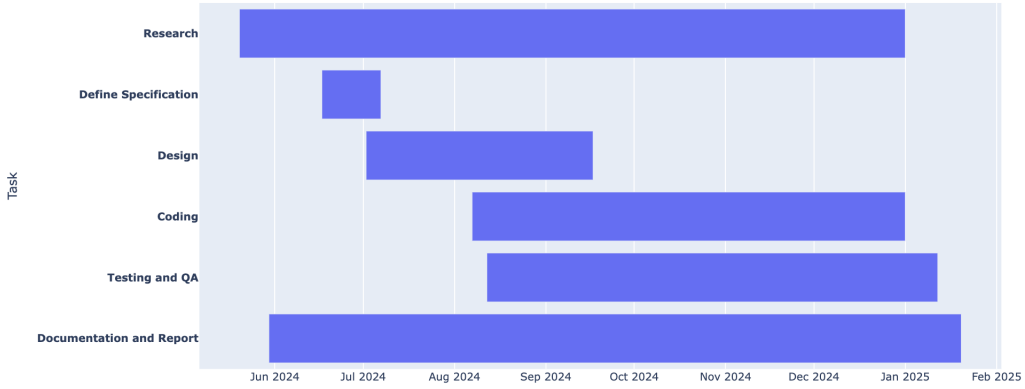

Figure 1.1: Gantt Chart

1.6 System Requirements

1.6.1 Development Requirements

Table 1.1: Development Requirements

Software Requirements	**Hardware Requirements**
Programming Language: Python, Dart, Javascript	RAM: >= 8 GB Megapixels
Design Tools: Figma, Canva, Draw.io	CPU: i5 10th
Libraries: Librosa, PyAudio, Pytorch	GPU: P100 (Recommended)
Framework: Flutter, Django RestFramework	Storage: >= 50 GB
IDEs: VSCode, Android Studio	

1.6.2 Deployment Requirements

Table 1.2: Deployment Requirements

Software Requirements	**Hardware Requirements**
Android: >= 10	RAM: > 4 GB
Read/Write FileSystem	Storage: >= 20 GB
Internet Accessibility	Recording Quality >= 256 Kbps, 48 KHz
Database: Sqlfite	

CHAPTER 2
LITERATURE REVIEW

This literature review explores the progression of methodologies and technologies in the field, with a particular focus on the use of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for audio-based bird species identification. The review also examines the challenges associated with dataset quality and diversity, and the innovative strategies employed to address these issues, providing a comprehensive overview of the current state of research and future directions in avian bioacoustics.

2.1 Related Works

BirdNET is a cutting-edge research platform developed through collaboration between the K. Lisa Yang Center for Conservation Bioacoustics at the Cornell Lab of Ornithology and the Chair of Media Informatics at Chemnitz University of Technology. Its primary aim is to detect and classify bird sounds using machine learning technologies, serving both experts and citizen scientists in their efforts to monitor and protect bird populations.

BirdNET can identify around 3,000 of the world's most common bird species, with plans to expand this number. Features such as a live submissions map and a Twitter bot are included to engage the community and share real-time data. The project is supported by donations and collaborations, offering opportunities for researchers and developers to contribute to its growth. BirdNET serves as an invaluable tool for bird enthusiasts, conservationists, and biologists alike, providing innovative solutions for large-scale acoustic monitoring and contributing to the conservation and understanding of avian biodiversity.

The BirdCLEF 2023 competition on Kaggle is a significant data science challenge that falls under the broader LifeCLEF initiative, aimed at pushing the boundaries of species identification and biodiversity monitoring through technological innovation. This par-

ticular competition focuses on the development of machine learning models that can identify bird species based on audio recordings. It presents a complex and realistic challenge due to the diversity of the audio recordings, which are collected from various environments and feature a wide range of bird species.

2.2 Related Research

The research paper ”Audio Classifier for Automatic Identification of Endangered Bird Species of Nepal” focuses on using deep learning techniques to identify endangered bird species from audio recordings. The dataset, collected from xeno-canto.org, comprises 2215 audio recordings of 41 bird species, 38 of which are endangered. This dataset was expanded to 6733 recordings through 10-second audio splitting and Gaussian noise augmentation, with 5407 recordings used for training, 639 for validation, and 687 for testing. The methodology involved handling imbalanced class distribution through data augmentation, employing Mel spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs) for feature extraction, and developing a custom Convolutional Neural Network (CNN) model and an EfficientNet model. The hyperparameters of these models were optimized using a genetic algorithm. The Mel spectrograms were created using Short-Time Fourier Transform, converting amplitudes to decibel scale, and applying Mel filter banks to the spectrograms. Similarly, MFCCs were derived by framing the audio signals, applying Discrete Fourier Transform, logarithmic scaling, Mel scaling, and Discrete Cosine Transform. The EfficientNet architecture utilized compound scaling for network depth, width, and resolution. The findings indicated that the proposed approach achieved satisfactory results in classifying the bird species. Model I, using Mel Spectrogram and EfficientNet, achieved an F1-score of 79%, while Model II, using Mel Spectrogram and Custom CNN, achieved 64%, and Model III, using MFCC and EfficientNet, achieved 72%. However, limitations include the relatively small dataset size and the need for further enhancement in model robustness and accuracy. [4]

The paper ‘Audio Bird Classification with Inception-v4 extended with Time and Time-Frequency Attention Mechanisms’ presents an innovative adaptation of the Inception-v4 deep convolutional network for bioacoustic classification, focusing specifically on

bird sound recognition. The datasets employed include various bird sounds, prominently from the BirdClef2017 challenge, consisting of 1500 bird species recordings. The methodology revolves around treating bird sound classification as an image classification problem through transfer learning. The Inception-v4 model, initially pretrained on ImageNet, was adapted to process time-frequency representations of bird sounds by converting these sounds into RGB images using three log-spectrograms generated via fast Fourier transform at different scales (128, 512, 2048 bins).The findings demonstrate that the model, termed 'Soundception', integrates time and time-frequency attention mechanisms effectively, significantly improving classification accuracy. The results highlight Soundception's outstanding performance, achieving a mean average precision (MAP) of 0.714 in classifying 1500 bird species, 0.616 MAP for background species, and 0.288 MAP for soundscapes with time-codes, making it the top model in the BirdClef2017 challenge across multiple tasks. However, limitations include the incomplete convergence of the model due to computational constraints and the extensive GPU resources required for training. The paper concludes with a discussion on future improvements, such as exploring different scalable optimizations and incorporating stacked GRU layers for better audio-to-image representation learning, underscoring the potential of transfer learning from advanced image classification models to acoustic domains.[5]

This research paper presents significant advancements in the field of 'Automatic bird species identification through the integration of audio signal processing and neural networks'. The study, conducted by Chandu B et al. outlines a robust methodology for identifying bird species from audio recordings, leveraging a combination of meticulously curated datasets and machine learning techniques. The dataset was manually compiled from both local recordings and online resources such as xeno-canto.org, which apart from bird songs also contains ambient noise and human voices to simulate real-world conditions. Pre-processing techniques including pre-emphasis, framing, silence removal, and reconstruction were applied to the audio clips to enhance the relevant frequency components and eliminate unnecessary noise, ensuring the purity of the dataset. Spectrograms of these processed clips were generated and used as input for

training a convolutional neural network (CNN), specifically AlexNet, chosen for its high accuracy in image classification tasks. Through transfer learning, AlexNet was adapted to recognize bird species from the spectrograms, achieving a classification accuracy of 97% in controlled environments. However, recognizing the variability of real-world conditions, the researchers retrained the model with datasets containing ambient noise, achieving a real-time classification accuracy of 91%. Despite these promising results, the study acknowledges limitations such as the relatively small size of the dataset and the need for further tuning of performance parameters to improve robustness[6].

The paper 'An Ensemble of Convolutional Neural Networks for Audio Classification' delves into a comprehensive study on CNN classification using different architectures, data augmentation techniques, and audio signal representations, aimed at enhancing audio classification tasks across various datasets. The study employs three datasets: BIRDZ, CAT, and ESC-50, each offering unique challenges in audio classification. The methodology involves training five convolutional neural networks (CNNs) with four audio representations combined with six different data augmentation methods, resulting in thirty-five subtypes of ensembles. The audio representations include techniques such as the Discrete Gabor Transform (DGT), Waveform Similarity OverLap Add (WSOLA), and Phase Vocoder. The data augmentation methods encompass procedures like short spectrogram augmentation, random time shift, and frequency masking. The CNN architectures are pre-trained models fine-tuned with these augmented datasets to boost classification accuracy. The findings reveal that the ensemble method outperforms standalone networks, achieving 97% accuracy on the BIRDZ dataset, 90.51% on the CAT dataset, and 88.65% on the ESC-50 dataset. The study also highlights that the best-performing CNNs are VGG16 and VGG19, with DGT as the most effective signal representation. However, the study acknowledges limitations, such as the computational cost of training ensembles and the variability in performance across different augmentation techniques[7].

The literature review focused on the 'Analysis of bird call datasets sourced from Xeno-Canto', comprising 72,172 samples from 264 bird species in 16-bit wav format with

a 16 kHz sampling rate. The methodology involved preprocessing the audio data to filter out low-frequency noise and normalize signal amplitude, followed by generating Mel-spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs) as inputs for deep learning models. The Mel-spectrograms were produced using discrete Fourier transform (DFT), and the MFCCs were derived by applying discrete cosine transform (DCT) to the Mel-spectrogram. The study employed various metrics to evaluate the performance of these methods, including ROC analysis to visualize model effectiveness. Findings indicated that the proposed models showed significant promise in identifying bird species from their calls, with improvements in classification accuracy compared to previous approaches. However, limitations were noted, including potential biases in the dataset due to uneven sample distribution across species and the challenge of background noise affecting signal quality. Future work suggested enhancing noise reduction techniques and exploring more sophisticated neural network architectures to further improve model robustness and accuracy.[8]

The study conducted an in-depth analysis of 'bird species recognition through acoustic monitoring', utilizing a robust dataset of bird sound samples, meticulously annotated and validated for accuracy. The dataset, referred to as SD, comprises multispecies bird sound recordings, each labeled with species name and sample ID, along with corresponding metadata, providing a comprehensive foundation for model training and evaluation. Methodologically, the research employed a spectrogram-based feature extraction approach, leveraging Short-Time Fourier Transform (STFT) to capture the intricate temporal and spectral characteristics of bird sounds. This was followed by the application of a Multilayer Perceptron (MLP) classifier to distinguish between different bird species. The findings reveal that the proposed model achieved high recognition accuracy, with some species being identified with perfect precision, recall, and accuracy (100%), though the performance varied across species, with a few showing lower recognition rates (86.9%) and precision/recall values ranging between 5075%. The results demonstrated an overall classification accuracy of 96%, with cross-validation accuracy standing at 81.4%, highlighting the model's robustness yet indicating room for improvement in generalizability across diverse datasets. Despite the promising re-

sults, the study acknowledges several limitations, including the variability in recognition accuracy among different species and the potential influence of environmental noise on model performance. Future work is suggested to explore feature and model fusion techniques, integrate the model with cloud-based systems for real-time recognition, and expand the dataset to include a broader range of bird species to enhance the model's applicability and accuracy in practical scenarios.[9]

The dataset used in this study comprises recordings labeled by species from California and Nevada, USA. It includes 91 species, with 30 audio samples per species, amounting to a total of 2,730 MP3 files. The methodology adopted involves three main steps: pre-processing, feature extraction, and deep learning modeling. For feature extraction, MFCCs were obtained using the Python library python_speech_features, with parameters such as sample rate, 13 cepstrum coefficients, 26 filterbank filters, and an FFT size of 512. Mel spectrograms were extracted using the Librosa library, employing parameters such as a sample rate, an FFT window size of 2048, a hop length of 512, and 128 Mel bands. In the deep learning modeling, CNNs and LSTMs were compared for their effectiveness in classifying bird sounds. CNNs demonstrated superior training accuracy with Mel spectrogram features, achieving 99.05% and 98.76% accuracy for 3-second and 1.5-second spectrograms, respectively. In contrast, LSTMs achieved lower training accuracies of 75.85% and 73.29% under similar conditions. These results highlight the superior ability of CNNs to leverage the spatial and frequency-related patterns in Mel spectrograms for accurate bird species classification.[10]

CHAPTER 3
METHODOLOGY

This chapters describes the bird species identification using audio.

3.1 Working Mechanism for Identification using Audio

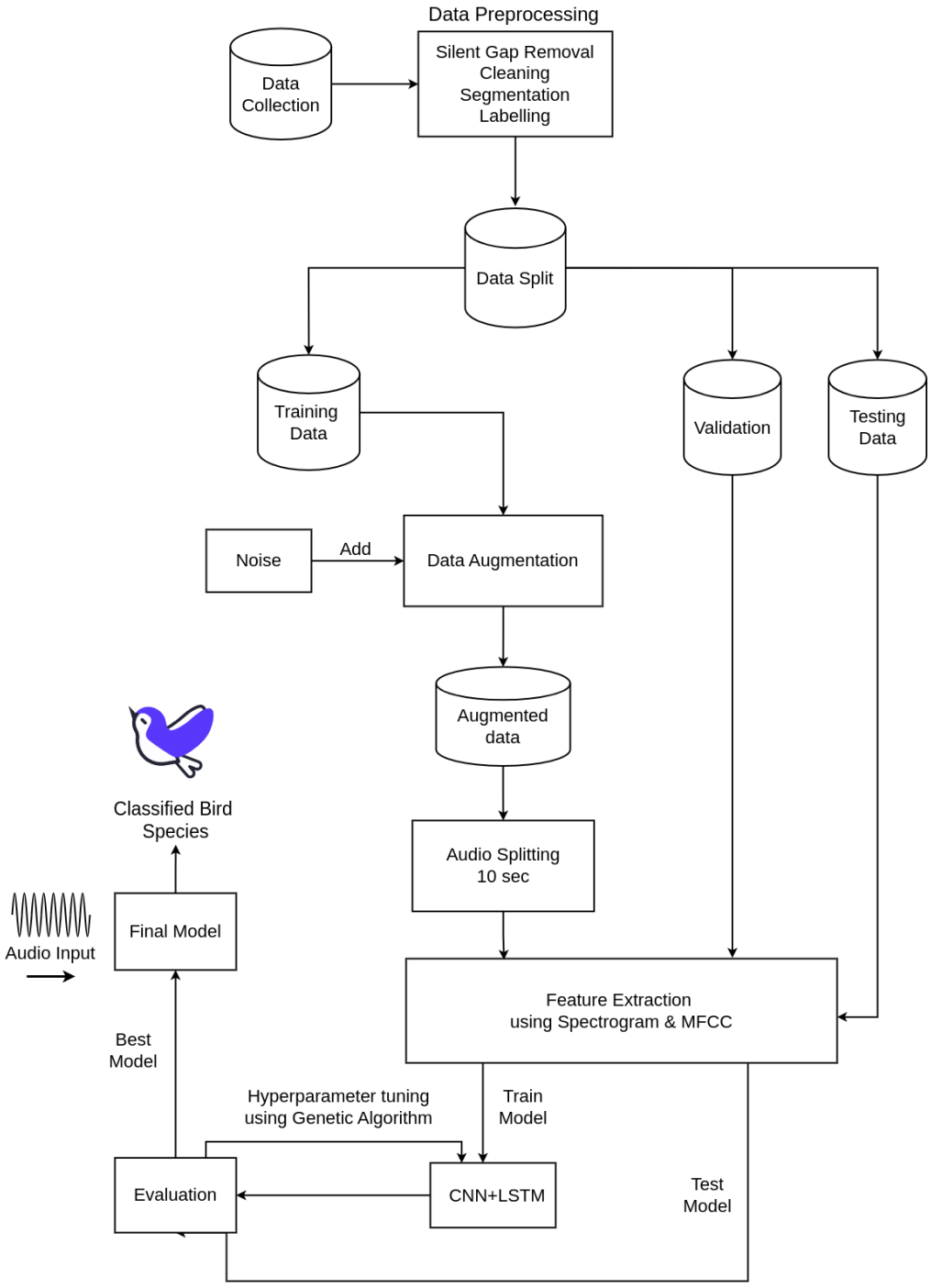

Figure 3.1: Block diagram for the working mechanism of the system

The working mechanism for bird identification from audio utilizes the CharaNET dataset, housing a diverse collection of avian vocalizations. Our methodology revolves around the utilization of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) architectures. The primary objective of this study is to achieve high levels of accuracy in identifying a broad spectrum of 41 distinct bird species. Here lies the detailed explaination of the methodology from working mechanism to model training.

3.1.1 Dataset

For this project, we will utilize the CharaNet dataset, which is specifically curated to include audio files of Nepal's endangered bird species. This dataset was collected from *xeno-canto.org*, a platform where bird sounds from around the world are shared by contributors who travel extensively to capture these sounds. From this source, we will gather 2215 audio recordings representing 41 bird species, 38 of which are listed as endangered in Nepal.

To augment the dataset, we propose employing a 10-second split method, which will expand the dataset to 6733 audio recordings. Additionally, for bird species with fewer than 30 recordings, we will add Gaussian noise to further increase the number of samples. This augmentation process is expected to result in a robust dataset comprising 5407 audio recordings for training, 639 for validation, and 687 for testing. This comprehensive dataset will ensure that our model can learn effectively and generalize well to new, unseen data.

This augmentation methodology and the CharaNet dataset have been referenced in prior research[4], demonstrating the efficacy of such an approach.

3.1.2 Dataset Overview for Audio

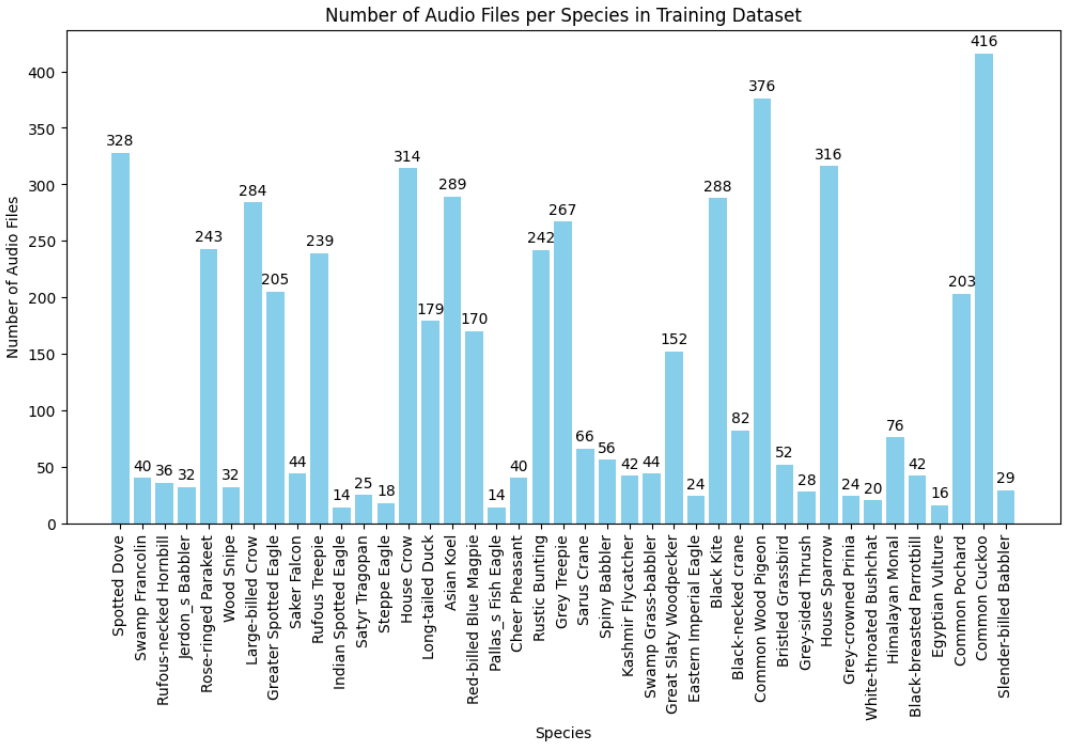

Figure 3.2: Training Dataset for audio

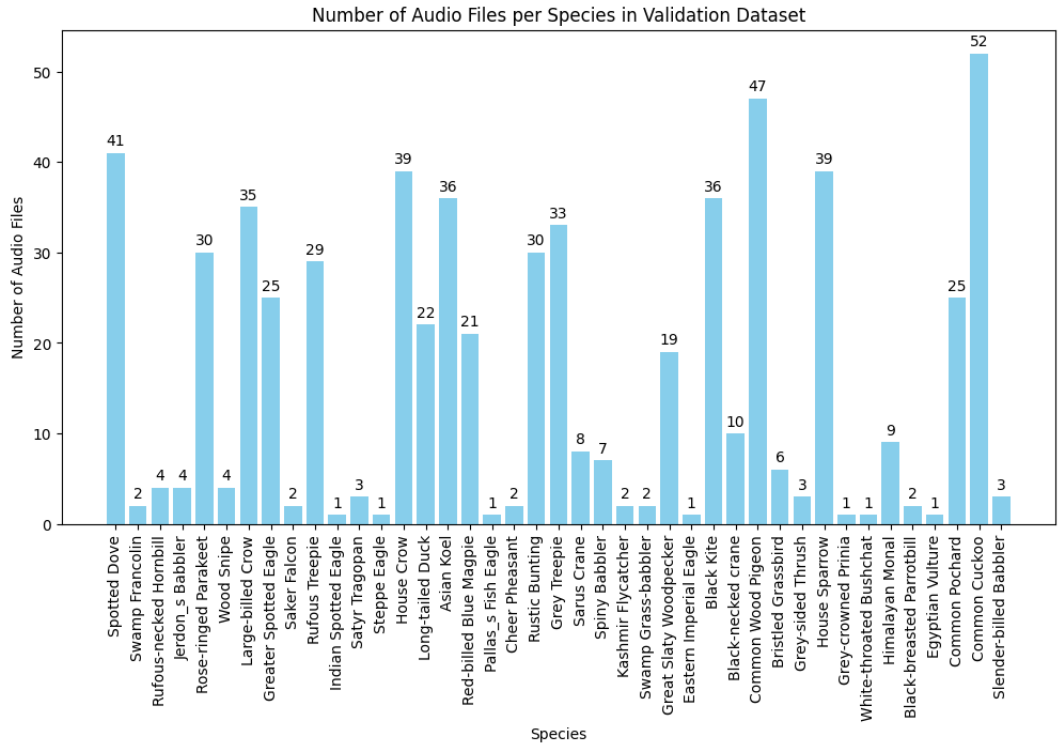

Figure 3.3: Validation Dataset for audio

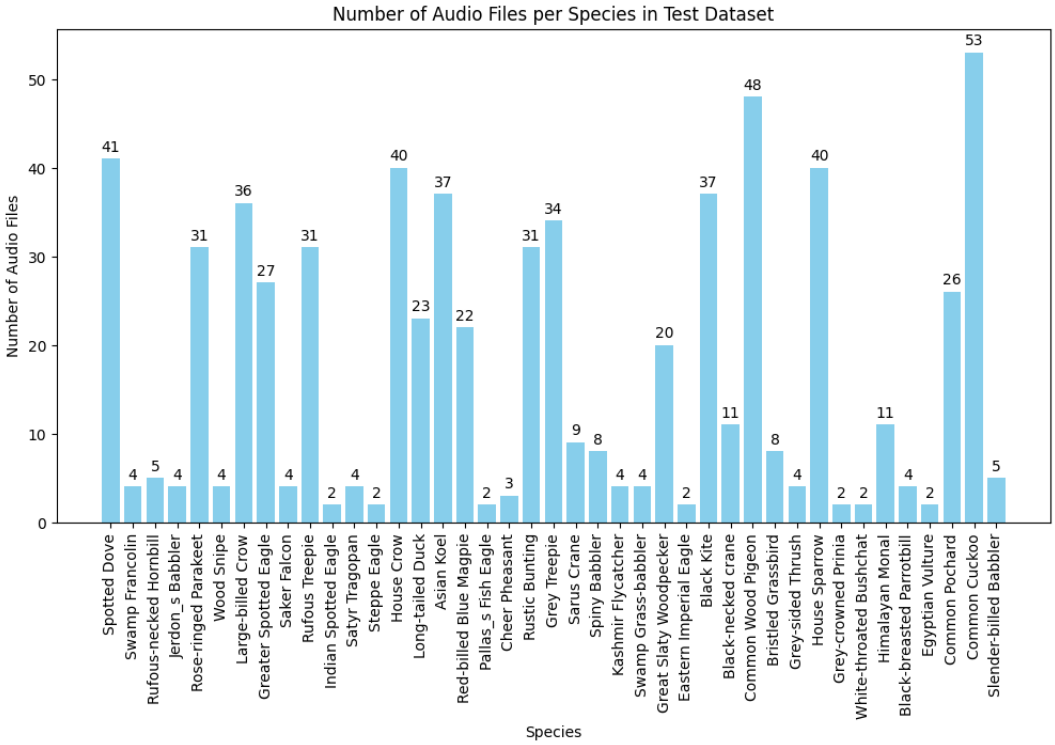

Figure 3.4: Testing Dataset for audio

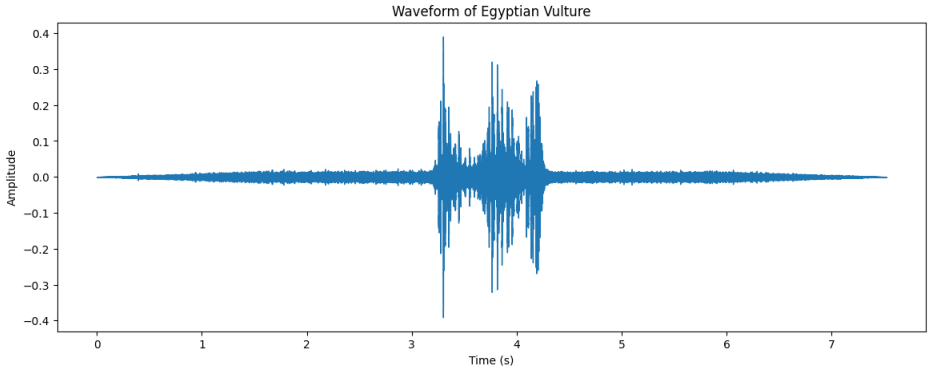

Figure 3.5: Sample of the Data

3.1.3 Data Preprocessing

The collected audio data undergoes thorough preprocessing to ensure its suitability for model training. This process includes:

- **Silence Gap Removal:** After observing some of the data samples, it was found that the audio has some silence gaps in either of the ends as seen in 3.5. These could be removed employing the silence removal algorithm. The processing starts from both ends and moves toward the center. In that processing, the local mean of the window segment of the audio wave is calculated and compared with the audio's global mean. If the local mean is smaller than the global mean, the window segment is c onsidered to contain insignificant data, thus the segment is clipped from the original audio[11]. Algorithm 3.1.12.1 clarifies this.
- **Segmentation:** Dividing the continuous audio recordings into smaller, more manageable segments, ensuring consistency in input length.
- **Cleaning:** Removing any background noise and irrelevant sounds that could interfere with the training process.
- **Labeling:** Assigning the correct bird species labels to each audio segment, which is crucial for supervised learning.

3.1.4 Data Splitting

After preprocessing, the data is split into three sets:

- **Training Data:** Used to train the machine learning models.
- **Testing Data:** Used to evaluate the model's performance during training.
- **Validation Data:** Used to validate the final model's performance and prevent overfitting.

This structured approach ensures that the model can learn effectively from the training data while being evaluated on unseen data to measure its generalization capabilities.

3.1.5 Data Augmentation

To further enhance the robustness of the model, data augmentation techniques are applied to the training data. This includes adding various types of noise to the audio segments to create more diverse training samples and prevent the model from overfitting to specific patterns in the data.

$$\text{Augmented Data} = \text{Original Data} + \text{Noise} \tag{3.1}$$

By introducing these variations, the model becomes more resilient to different audio conditions and better at generalizing to new recordings.

3.1.6 Audio Splitting

The augmented audio data is then split into smaller 10-second clips. This standardization ensures that the input size remains consistent, which is essential for the feature extraction and modeling stages.

3.1.7 Feature Extraction

Feature extraction is a critical step where the audio clips are transformed into a format that can be fed into machine learning models. We use two primary techniques for feature extraction: Spectrograms and Mel-Frequency Cepstral Coefficients (MFCC).

3.1.7.1 Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies in a sound signal as they vary with time. It is generated by applying the Short-Time Fourier Transform (STFT) to the audio signal. This transformation provides insight into how the frequency

content of the signal changes over time.

$$X(n) = \sum_{m=0}^{N-1} x(m) \cdot w(n-m) \tag{3.2}$$

where $x(m)$ is the audio signal and $w(n)$ is the window function.

3.1.7.2 Mel-Frequency Cepstral Coefficients (MFCC)

MFCCs are coefficients that collectively describe the short-term power spectrum of a sound signal. The process of obtaining MFCCs involves several steps:

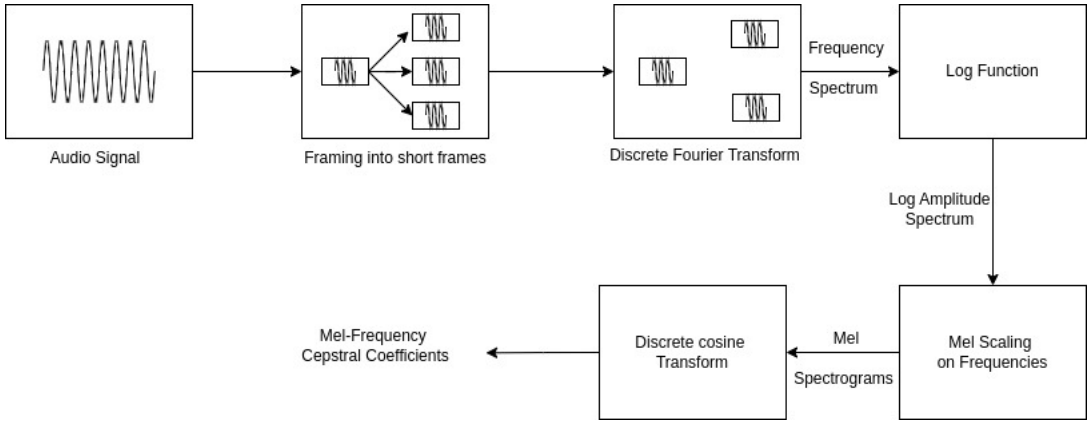

Figure 3.6: Feature Extraction Using Spectrogram and MFCC

1. **Framing:** Divide the audio signal into short frames.
2. **Discrete Fourier Transform (DFT):** Convert each frame to the frequency domain.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn} \tag{3.3}$$

3. **Log Function:** Apply a logarithm to the amplitude spectrum.

$$S_{\log}(k) = \log(|X(k)|) \tag{3.4}$$

4. **Mel-Scaling:** Map the frequencies to the Mel scale, which better represents how

humans perceive sound.

$$f_{\text{mel}} = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \tag{3.5}$$

5. **Discrete Cosine Transform (DCT):** Convert the Mel spectrum to the cepstral domain, yielding the MFCC features.

$$C(n) = \sum_{k=0}^{K-1} S_{\text{mel}}(k) \cdot \cos\left(\frac{\pi n(k+0.5)}{K}\right) \tag{3.6}$$

3.1.8 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are highly effective for extracting spatial features from spectrograms, making them well-suited for audio classification tasks. CNNs use convolutional layers to detect patterns and features in the input data by applying convolutional filters across the input spectrogram.

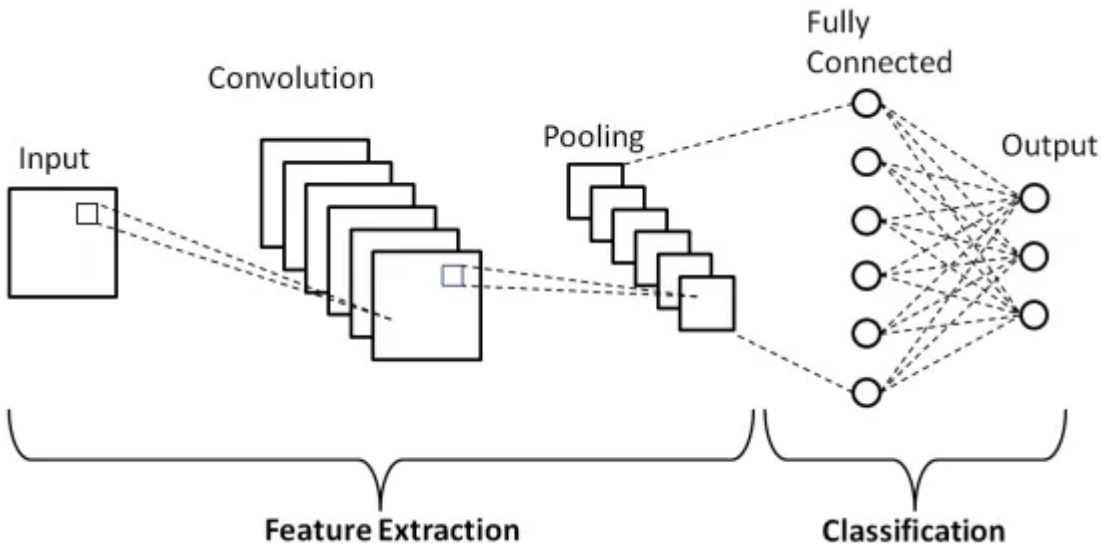

Figure 3.7: CNN architecture

The basic operations involved in a CNN include:

- **Convolution:** This operation involves applying a set of learnable filters (kernels)

across the input spectrogram to produce feature maps.

$$(f * x)(t) = \sum_{\tau=-\infty}^{\infty} x(\tau) \cdot f(t - \tau) \tag{3.7}$$

- **Activation Function:** The rectified linear unit (ReLU) activation function is applied to introduce non-linearity into the model.

$$f(x) = \max(0, x) \tag{3.8}$$

- **Pooling:** Pooling layers reduce the spatial dimensions of the feature maps, typically using max pooling to retain the most significant features.
- **Fully Connected Layers:** After several convolutional and pooling layers, the feature maps are flattened and passed through fully connected layers to produce the final classification output.

3.1.9 Long Short-Term Memory Networks (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) capable of capturing temporal dependencies in sequential data. This makes them well-suited for processing audio signals where the order of data points matters. LSTMs are designed to overcome the limitations of traditional RNNs by addressing the vanishing gradient problem through the use of gates that regulate the flow of information.

The key components of an LSTM cell include:

- **Forget Gate:** Determines which information from the previous cell state should be discarded.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.9}$$

- **Input Gate:** Decides which new information should be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.10}$$

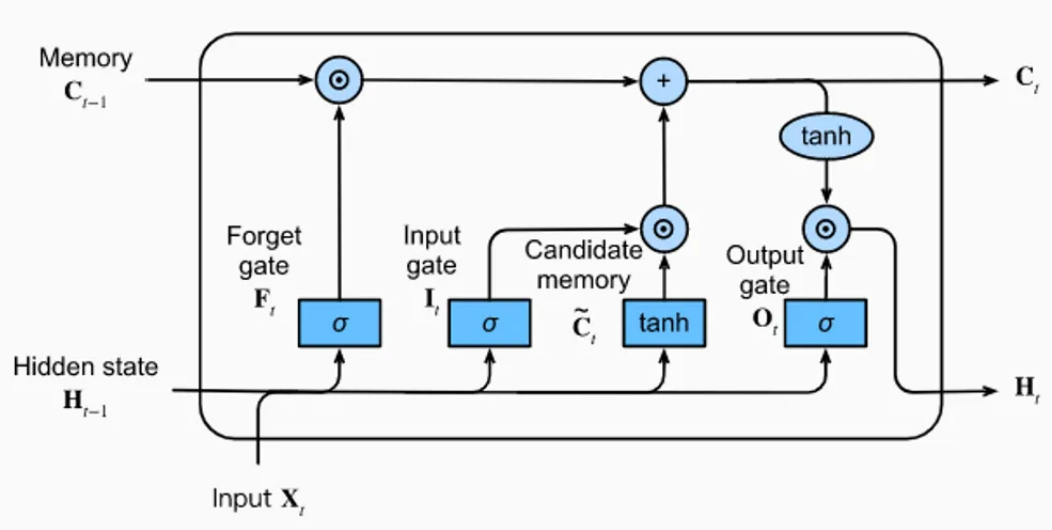

Figure 3.8: LSTM architecture

- **Candidate Cell State:** Creates new candidate values that could be added to the cell state.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.11}$$

- **Cell State:** The cell state is updated based on the input gate and the forget gate.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3.12}$$

- **Output Gate:** Determines the output of the LSTM cell.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.13}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{3.14}$$

3.1.10 Combined CNN and LSTM (CNN+LSTM)

A hybrid model that leverages the strengths of both CNN and LSTM architectures is developed. In this combined model, the CNN processes spectrograms to extract spatial features, which are then fed into LSTM layers to capture temporal patterns. The integration of these two models aims to utilize both spatial and temporal information,

thereby improving the overall classification accuracy. Fully connected layers at the end perform the final classification based on the features extracted by both CNN and LSTM components.

3.1.11 Hyperparameter Optimization using Genetic Algorithm

Genetic algorithms (GAs) are a powerful method for optimizing hyperparameters in machine learning models. Genetic Algorithm have proven to significantly improve the performance metrics of the CNN model instead of using hand tuned approach for hyperparameters. This section outlines the steps involved in using GAs for hyperparameter optimization[12].

- **Encoding the Hyperparameters**
  - Hyperparameters are represented as a chromosome, where each hyperparameter is a gene in the chromosome.
  - For example, in a neural network, a chromosome might include genes for the learning rate, number of layers, number of neurons per layer, and activation functions.
- **Initial Population**
  - An initial population of chromosomes is generated randomly, with each chromosome representing a different set of hyperparameters.
- **Fitness Function**
  - A fitness function is defined to evaluate the performance of each set of hyperparameters.
  - This typically involves training the model with the given hyperparameters and measuring its performance on a validation set.
- **Selection**
  - Selection involves choosing the best-performing chromosomes to serve as parents for the next generation.
  - Various selection methods can be employed, such as tournament selection, roulette wheel selection, or rank-based selection.

- **Crossover (Recombination)**
  - Crossover combines pairs of parent chromosomes to produce offspring for the next generation.
  - This is done by swapping segments of parent chromosomes to create new chromosomes, thereby combining features of both parents.
- **Mutation**
  - Mutation introduces random changes to some of the genes in the offspring chromosomes.
  - This helps maintain genetic diversity in the population and allows the algorithm to explore a broader search space.
- **Replacement**
  - The current population is partially or entirely replaced with the new generation of chromosomes, ensuring that better solutions are carried forward while allowing for exploration of new possibilities.
- **Termination**
  - The process of selection, crossover, mutation, and replacement is repeated until a termination criterion is met.
  - This could be a set number of generations, convergence of fitness scores, or achieving a satisfactory performance level.
- **Best Solution**
  - The best chromosome at the end of the process represents the optimal or near-optimal set of hyperparameters for the model.

3.1.12 Algorithms Used

3.1.12.1 Silent Gaps Removal

Algorithm 1 Clipping of silent gaps from both ends

wav $\leftarrow$ sampled audio signal
$\Delta \leftarrow$ appropriate window length
/* In our code, $\Delta = 500$ for 16KHz sampling rate */
INPUT: wav, Δ
PROCESS:
wavAvg $\leftarrow$ Average($|$wav$|$)
$N \leftarrow$ Length(wav)
/* Removing the silent gap from the start */
for idx $= 0, \Delta, 2\Delta, \ldots, N - \Delta$ **do**
    win $\leftarrow$ wav[idx : idx $+ \Delta$]
    winAvg $\leftarrow$ Average($|$win$|$)
    if winAvg $>$ wavAvg **then**
        wav $\leftarrow$ wav[idx :]
        break
    end if
end for
/* Removing the silent gap from the end */
for idx $= N - \Delta, N - 2\Delta, \ldots, 0$ **do**
    win $\leftarrow$ wav[idx : idx $+ \Delta$]
    winAvg $\leftarrow$ Average($|$win$|$)
    if winAvg $>$ wavAvg **then**
        wav $\leftarrow$ wav[: idx]
        break
    end if
end for
OUTPUT: processed_wav $\leftarrow$ wav

3.1.12.2 Genetic Algorithm

Algorithm 2 Genetic Algorithm for Hyperparameter Optimization

Initialize the population with random hyperparameters.
for generation = 1 to N **do**
    Evaluate the fitness of each individual in the population.
    Select individuals to be parents based on their fitness scores.
    Generate offspring through crossover.
    Apply mutation to the offspring.
    Replace the old population with the new generation.
end for
Return the best solution found.

3.1.12.3 Fitness Function

The fitness function evaluates the performance of hyperparameters by training and validating the model:

Algorithm 3 Fitness Function

1: Train the model with given hyperparameters.
2: Evaluate the model's performance on a validation set.
3: **return** Performance metric (e.g., accuracy, F1 score).

3.1.12.4 Mel Spectrogram

Algorithm 4 Mel Spectrogram Extraction

1: **Input:** Audio signal $x(t)$.
2: **Output:** Mel spectrogram S_{mel}.
3: Apply STFT to generate spectrogram $S(f, t)$.
4: Convert amplitudes of $S(f, t)$ to dB scale, obtaining $S_{\text{dB}}(f, t)$.
5: **Convert frequencies to Mel scale**.
6: Choose number of mel bands N_{mel}.
7: **Construct mel filter banks**.
8: Convert f_{min} and f_{max} of $S_{\text{dB}}(f, t)$ to Mel scale.
9: Divide Mel scale range into N_{mel} intervals.
10: Convert center frequencies of Mel bands back to Hertz.
11: Round center frequencies to nearest bins.
12: Design triangular band pass filters for each Mel band.
13: Apply mel filter banks to $S_{\text{dB}}(f, t)$ to obtain S_{mel}.

3.1.12.5 Mel-Frequency Cepstral Coefficients Extraction

Algorithm 5 MFCC Extraction

1: **Input:** Audio signal $x(t)$
2: **Output:** MFCC coefficients.
3: Frame the signal into short frames.
4: Apply a window function to the frames.
5: Apply DFT to generate the frequency spectrum of each frame.
6: Apply logarithm to the spectrum to get log amplitude spectrum.
7: Perform Mel scaling using filter banks to get Mel spectrogram.
8: Apply DCT to the Mel spectrogram to get MFCCs.

3.2 Mapping Location of Bird in Map

Mapping the location of a bird in an application using **google_maps_flutter** and **geolocator** involves using the classified bird data to pinpoint its location on a map. This is particularly useful for bird watchers and researchers to track bird sightings.

Algorithm 6 Mapping Location of Bird in Map

1: Initialize the Google Maps widget
2: Capture the image of the bird
3: Get the classified bird species and captured location.
4: Convert the location data to latitude and longitude coordinates
5: Add a marker to the map at the specified coordinates
6: Update the map view to center on the new marker

3.3 System Diagram

The usecase diagram for Bird species identification from Audio and Image is given below:

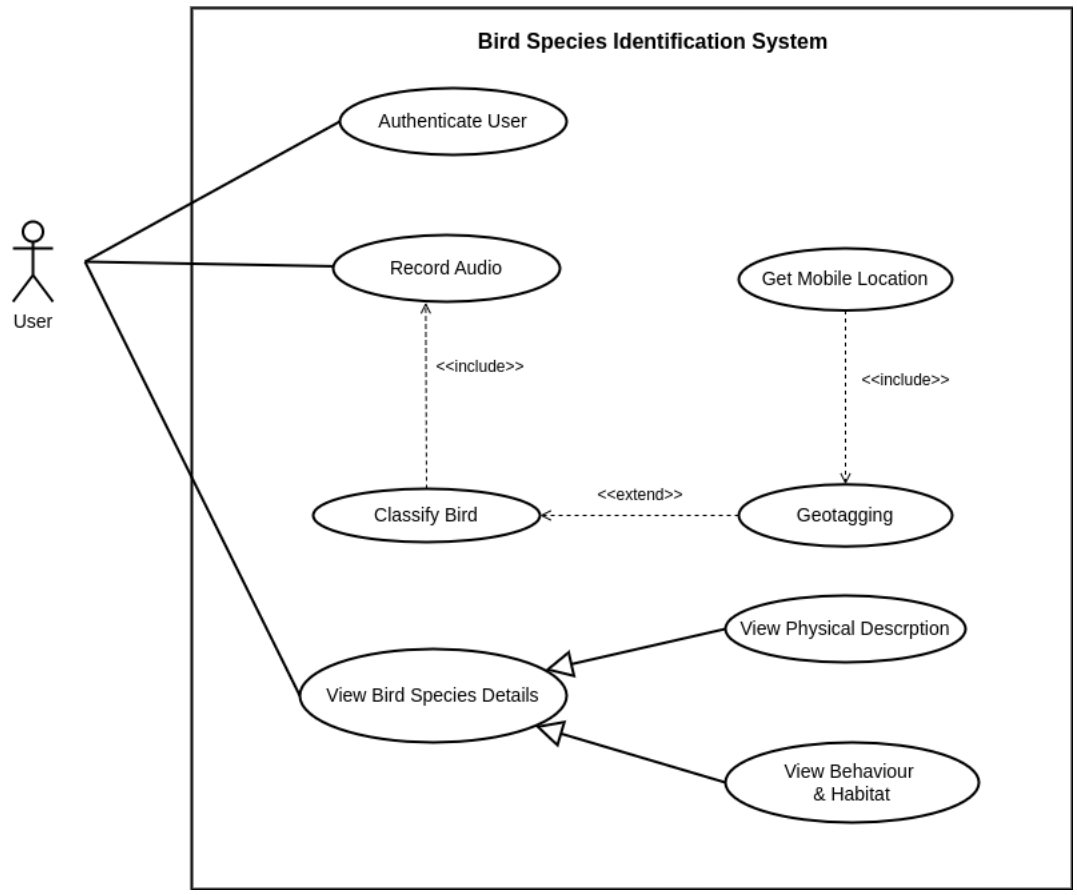

Figure 3.9: Usecase Diagram for FeatherFind

3.4 Software Development Model

This project will be using an incremental methodology since it offers a functioning prototype at an early stage of development. The requirements and scope of the project can be altered as necessary by studying the prototype. The rationale for the preference for this software development strategy is the flexibility offered by adopting the incremental technique. In this paradigm, the project goes through several releases or iterations prior to its official release.

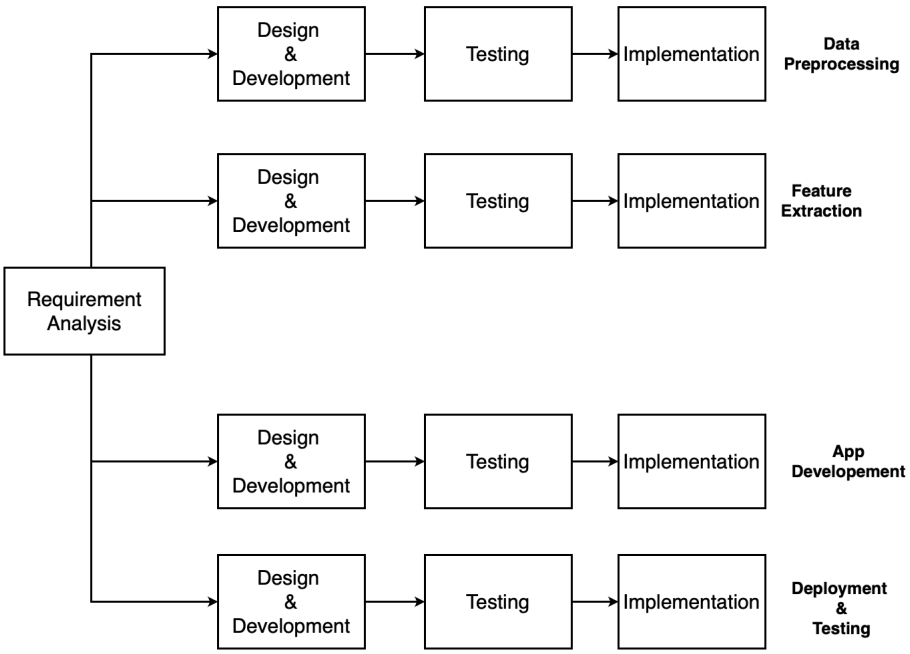

Figure 3.10: Incremental Model for development of FeatherFind

CHAPTER 4
EPILOGUE

4.1 Work Done

In the course of our project on Bird Species Classification from Audio, we have accomplished several key tasks to progress our analysis and model development:

4.1.1 Data Collection

We curated a dataset of bird calls for 41 bird species found in Nepal using Xeno-Canto, a globally recognized repository of bird sound recordings. This platform provided diverse audio recordings for the selected bird species, forming the foundation of our dataset.

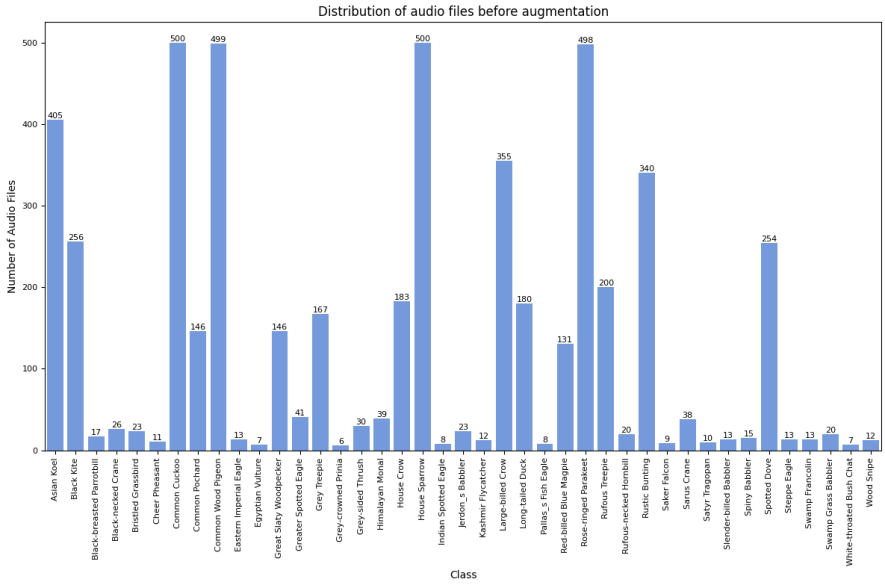

Figure 4.1: Data before performing augmentation

4.1.2 Data Augmentation

Initial analysis of the collected data revealed significant class imbalance, with some species having over 400 recordings while others had fewer than 50. Additionally, the audio recordings varied in length. To address these issues:

- **Audio Clipping:**
  - Recordings were clipped into segments of 10 seconds each.
  - Clips shorter than 5 seconds were discarded to avoid blank audio segments and insufficient data.
  - Clips between 5 and 10 seconds were padded with silence at the end to standardize their length to 10 seconds.
- **Augmentation Techniques:** To balance the dataset, augmentation techniques such as time stretching, phase shifting, and noise addition were applied. The parameters were varied to ensure diversity in the augmented data.
  - Each bird class was augmented to contain exactly 500 audio clips.

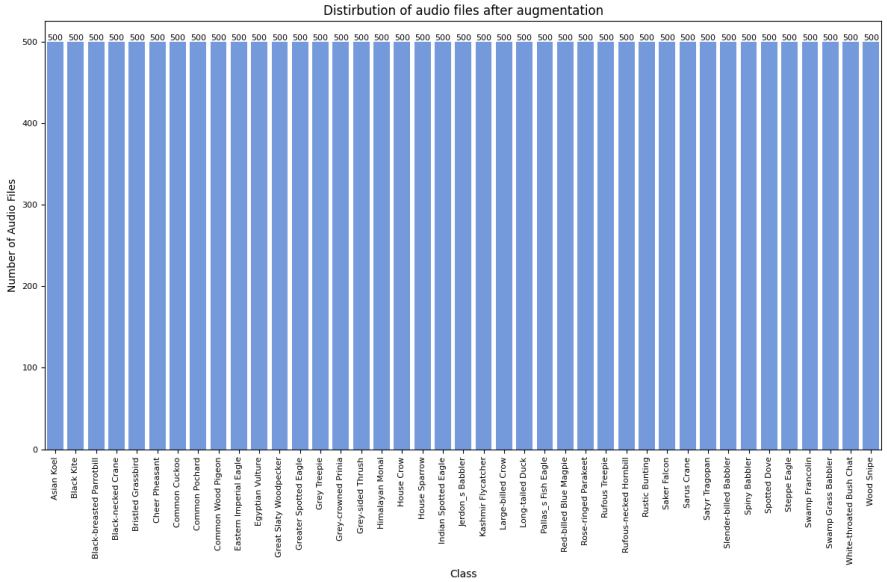

Figure 4.2: Data after performing augmentation

4.1.3 Data Preprocessing

The audio recordings were transformed into Mel Spectrograms for further analysis.

- **Conversion Details:**
  - Audio files were converted using a sample rate of 32,000 Hz.
  - The spectrograms were generated with a Hanning window and 48 Mel bands.
- **Dataset Expansion:**
  - Each of the 500 audio files per bird class was converted into corresponding Mel Spectrogram images.
  - The resulting dataset was divided into training, validation, and testing sets for model development.

4.1.4 Model Training

The classification model was built using **EfficientNetB3** as the base architecture.

- **Input Preprocessing:**
  - Mel Spectrogram images were resized to $128 \times 128 \times 3$ and normalized using the preprocess_input function of EfficientNet.
- **Regularization:**
  - Dropout layers and kernel regularizers were employed to mitigate overfitting.
- **Callback Functions:**
  - Early stopping, learning rate reduction on plateau, and checkpoint saving mechanisms were implemented to enhance training efficiency and performance.
- **Performance:**
  - On the test set, the model achieved an accuracy of **89.36%** with a cross-entropy loss of **0.42**.

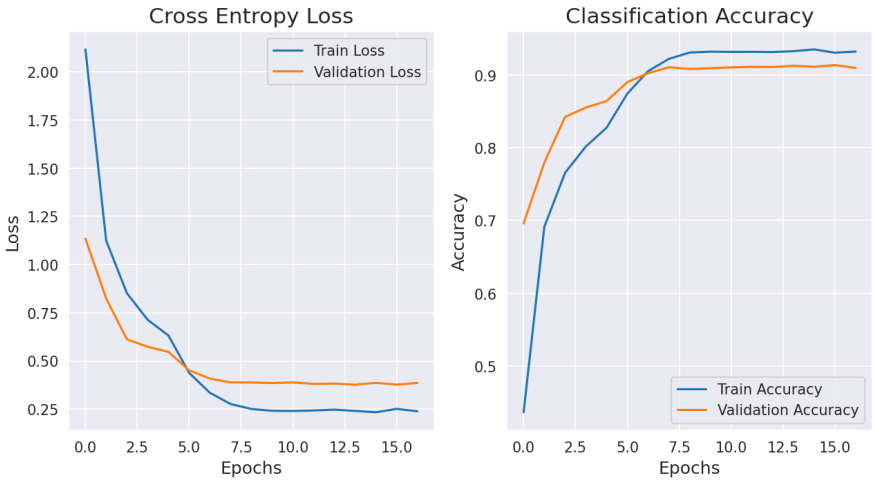

Figure 4.3: Accuracy and Cross-entropy loss for train and validation sets

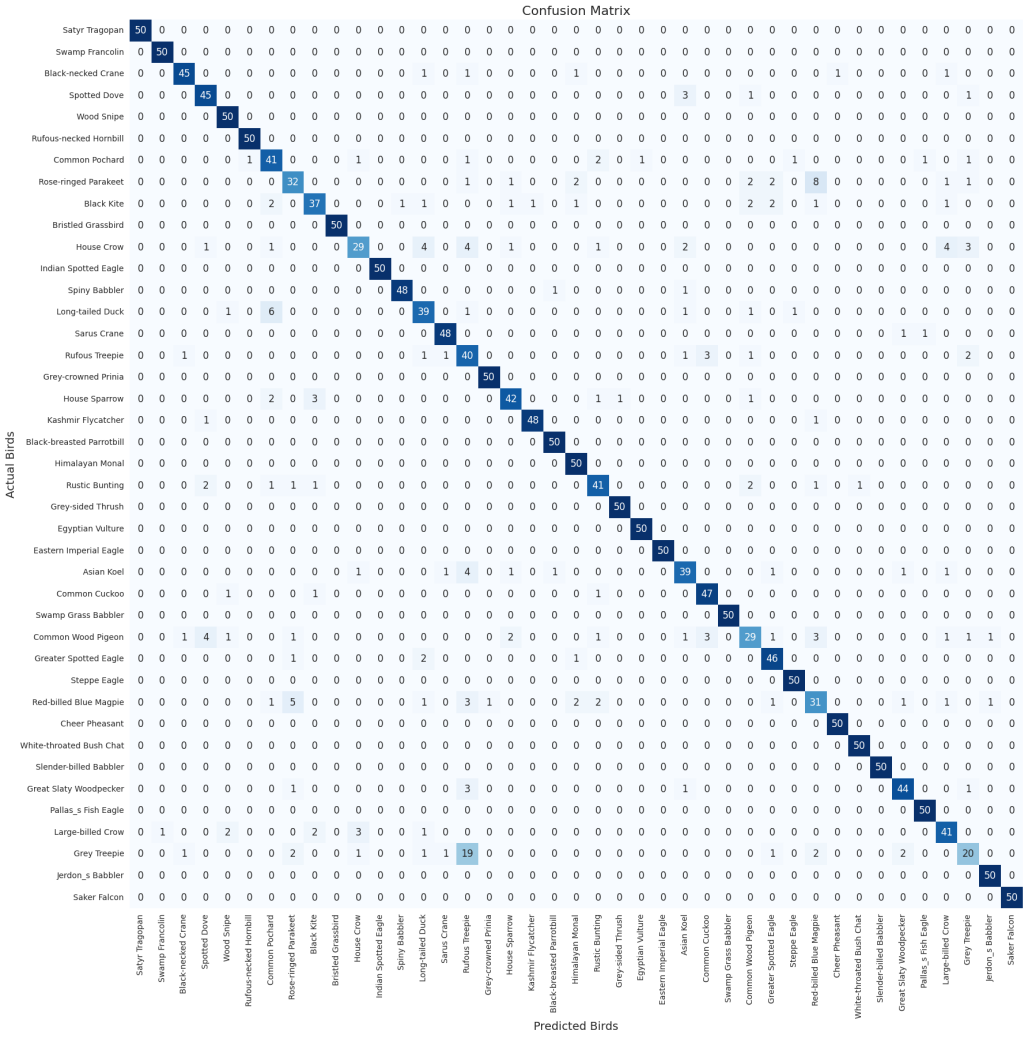

Figure 4.4: Confusion matrix for the model on test set.

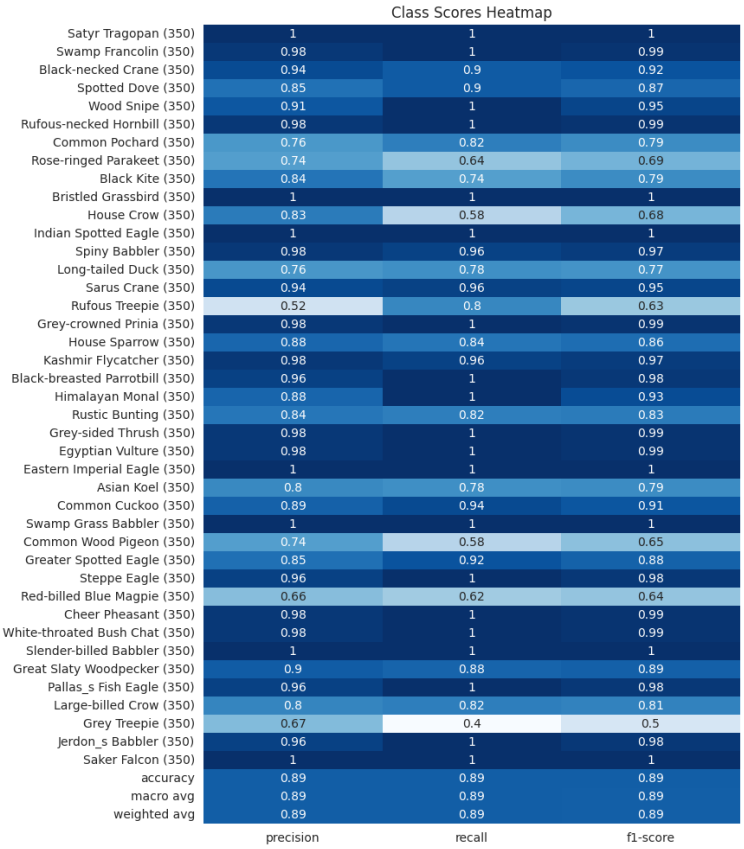

Figure 4.5: Classification report of the model.

4.1.5 Deployment to Huggingface spaces

A FastAPI application was developed using the trained model and deployed on **Huggingface spaces**. This deployment enables seamless API integration, allowing users to interact with the model for predictions efficiently and reliably.

4.1.6 Bird Sound Detection Model Training

To enhance the reliability of the species classification system, we incorporated a Bird Sound Detection model as a preliminary step to verify the presence of bird sounds in recorded audio. This approach is inspired by Lasseck (2018) [13], which demonstrated the effectiveness of using Deep Convolutional Neural Networks (DCNNs) for acoustic bird detection, achieving an AUC score exceeding **93.7%** on unseen data. Following this methodology, we employed a pre-trained InceptionV3 network for our binary classification task, achieving an initial AUC score of **83%**.

Dataset used for the training of the Bird Sound Detection Model is Field recordings, worldwide (”freefield1010”) - a collection of 7,690 excerpts from field recordings around the world, gathered by the FreeSound project, and then standardised for research. This collection is very diverse in location and environment, and for the BAD Challenge we have annotated it for the presence/absence of birds.

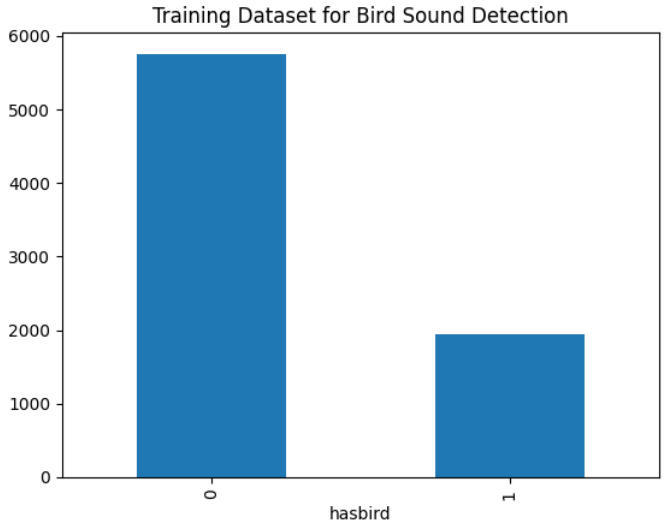

Figure 4.6: Dataset distribution for freefield1010

Dataset used for the evaluating the model’s performance is Crowdsourced dataset, UK (”warblrb10k”) - 8,000 smartphone audio recordings from around the UK, crowdsourced by users of Warblr the bird recognition app. The audio covers a wide distribution of UK locations and environments, and includes weather noise, traffic noise, human speech and even human bird imitations.

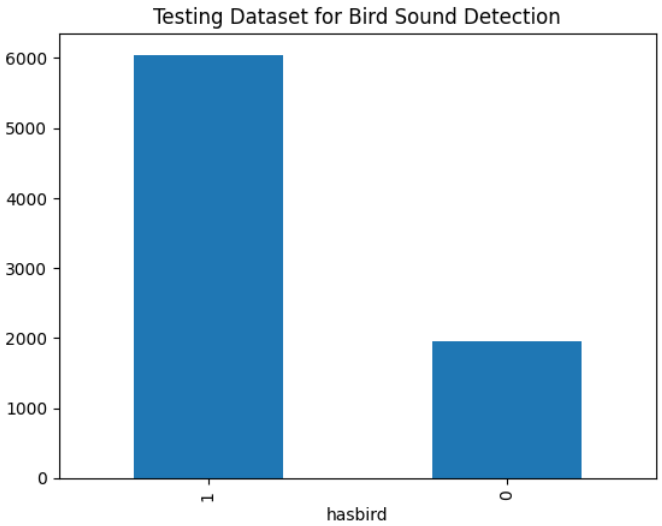

Figure 4.7: Dataset distribution for warblrb10k

After the model was subjected to the testing dataset, the following confusion matrix was obtained, yielding an accuracy of **87.28%**.

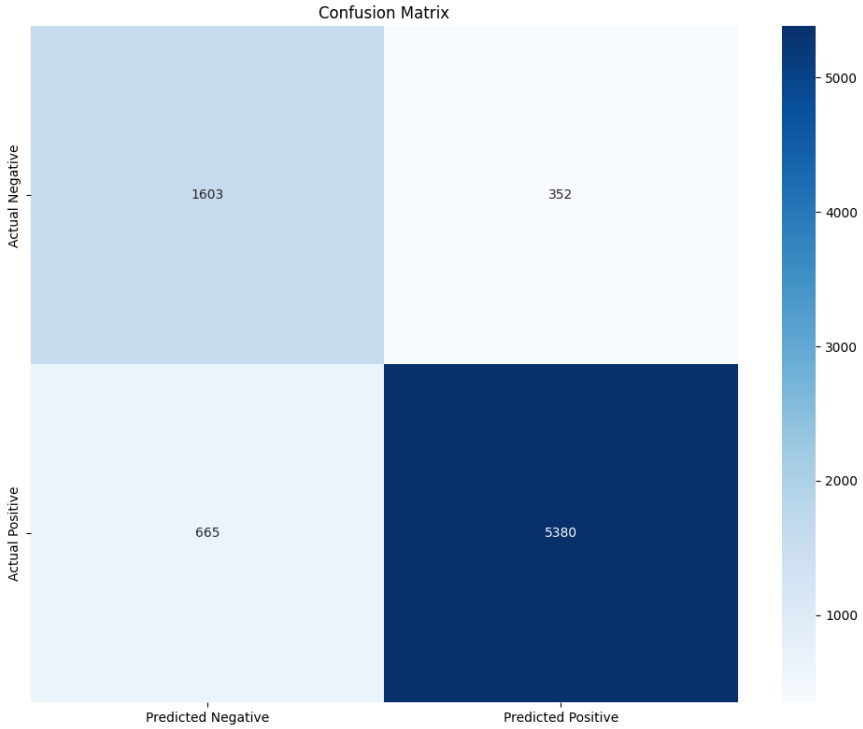

Figure 4.8: Confusion Matrix for Bird Sound Detection Model

To assess the effectiveness of our model in distinguishing between the target classes, we analyzed its performance using the Receiver Operating Characteristic (ROC) curve. The ROC curve provides a comprehensive view of the trade-off between the true positive

rate (TPR) and the false positive rate (FPR) at various threshold settings. The Area Under the Curve (AUC) metric, derived from the ROC curve, quantifies the overall discriminatory power of the model, with a higher AUC indicating better performance.

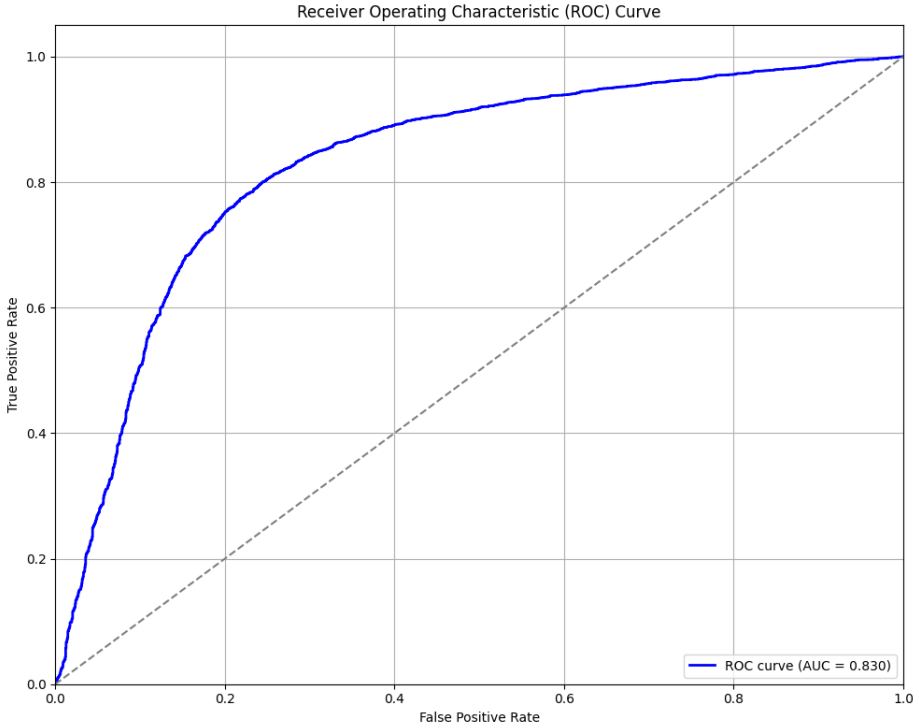

Figure 4.9: ROC Curve of the detection model

While our results are promising and validate the model's applicability, the specific preprocessing and data augmentation techniques outlined in the reference study, such as time and frequency stretching and the use of a high-pass filter, were not implemented. Incorporating these methods in future iterations could further enhance the model's performance and generalization to diverse recording conditions.

4.1.7 Mobile App Development

The mobile application allows the user to record the audio, visualize it with waveforms during recording, before sending the actual audio data to the server for processing. The user can also use the application to add their current location to the map.

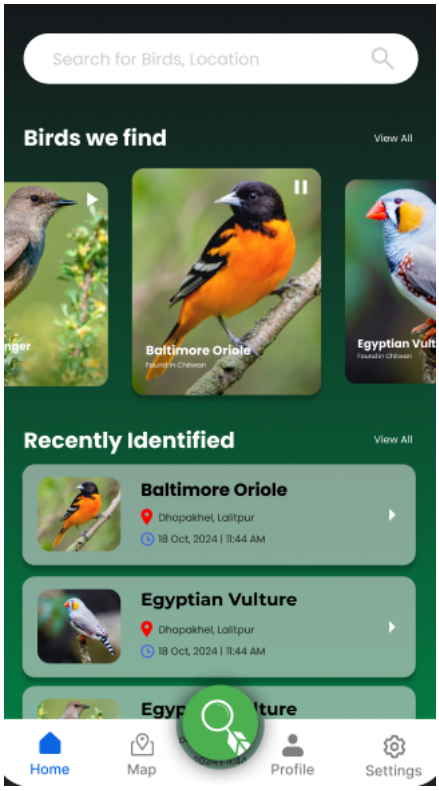

Figure 4.10: Home Page of FeatherFind.

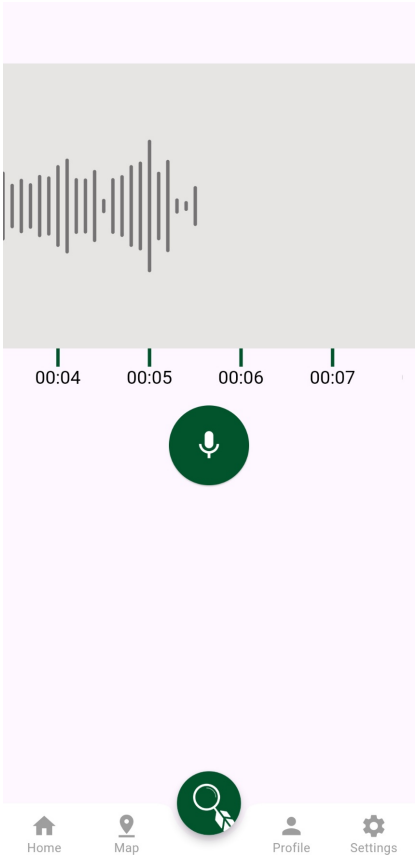

Figure 4.11: Audio Recording using FeatherFind.

4.2 Work Remaining

While substantial progress has been made in the Bird Species Classification from Audio project, several critical tasks remain to be addressed to further refine and enhance the system:

4.2.1 Model Training with CNN+LSTM

We will extend our model training to include a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network. The combination of CNNs and LSTMs is expected to capture both spatial and temporal features in the audio data more effectively, potentially leading to improved classification accuracy. We will experiment with various architectures and configurations to identify the most effective setup for our classification task.

4.2.2 Genetic Algorithm for Hyperparameter Tuning

To optimize the performance of our model, we plan to use a genetic algorithm for hyperparameter tuning. This approach will help us systematically explore and identify the most effective hyperparameter settings for our model, including learning rates, batch sizes, and network architecture parameters. The genetic algorithm will enable a more efficient and comprehensive search for optimal configurations.

4.2.3 Mobile App Integration

Once the model is properly trained and optimized, the next step will be to integrate the machine learning model into a mobile application. The mobile app will capture audio input from the user, preprocess it (e.g., feature extraction), and then feed it into the deployed model for classification.

These remaining tasks are essential to advancing the project and achieving our goal of an accurate and reliable bird species classification system from audio data.

REFERENCES

[1] I. O. Union, "Ioc world bird list," https://www.worldbirdnames.org/new/updates/, 2024, accessed: 2024-06-04.

[2] H. Nature, "National red list of nepal's birds," https://www.himalayannature.org/works/projects/national-red-list-of-nepals-birds/, 2024, accessed: 2024-06-03.

[3] C. Inskipp, H. S. Baral, T. Inskipp, A. P. Khatiwada, M. P. Khatiwada, L. P. Poudyal, and R. Amin, "Nepal's national red list of birds," *Journal of Threatened Taxa*, vol. 9, no. 1, pp. 9700–9722, 2017.

[4] R. Gautam, B. Khatiwada, B. P. Subedi, N. Duwal, and K. C. Dahal, "Audio classifier for automatic identification of endangered bird species of nepal," 2023.

[5] A. Sevilla and H. Glotin, "Audio bird classification with inception-v4 extended with time and time-frequency attention mechanisms." *CLEF (Working Notes)*, vol. 1866, pp. 1–8, 2017.

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

[7] L. Nanni, G. Maguolo, S. Brahnam, and M. Paci, "An ensemble of convolutional neural networks for audio classification," *Applied Sciences*, vol. 11, no. 13, p. 5796, 2021.

[8] H. Wang, Y. Xu, Y. Yu, Y. Lin, and J. Ran, "An efficient model for a vast number of bird species identification based on acoustic features," *Animals*, vol. 12, no. 18, p. 2434, 2022.

[9] R. Pahuja and A. Kumar, "Sound-spectrogram based automatic bird species recognition using mlp classifier," *Applied Acoustics*, vol. 180, p. 108077, 2021.

[10] S. Carvalho and E. F. Gomes, "Automatic classification of bird sounds: using mfcc and mel spectrogram features with deep learning," *Vietnam Journal of Computer Science*, vol. 10, no. 01, pp. 39–54, 2023.

[11] M. Dhakal, A. Chhetri, A. K. Gupta, P. Lamichhane, S. Pandey, and S. Shakya, "Automatic speech recognition for the nepali language using cnn, bidirectional lstm and resnet," pp. 515–521, 2022.

[12] A. Reiling, W. Mitchell, S. Westberg, E. Balster, and T. Taha, "Cnn optimization with a genetic algorithm," in *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, 2019, pp. 340–344.

[13] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks." in *DCASE*, 2018, pp. 143–147.