Bird Species Classification Using Vision Transformers

Submitted In Partial Fulfillment of Requirements
For the Degree Of

Honours in Data Science and Analytics
(Offered by Department of Computer Engineering)

By

Sakshi Borade
Roll No: 16010121026
Atharv Chandane
Roll No: 16010121030
Kunal Chaturvedi
Roll No: 16010121032
Sahil Chauhan
Roll No: 16010121034
Vedant Chavan
Roll No: 16010121035

Guide
Prof. Kirti Mishra

Somaiya Vidyavihar University Vidyavihar,
Mumabi - 400 077
2021-25

SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya Vidyavihar University
K. J. Somaiya College of Engineering

Certificate

This is to certify that the dissertation report entitled **Bird Species Classification Using Vision Transformers** submitted by Sakshi Borade(16010121026), Atharv Chandane(16010121030), Kunal Chaturvedi(16010121032), Sahil Chauhan(16010121034), and Vedant Chavan(16010121035) at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree Honours in Data Science and Analytics (**Offered by Department of Computer Engineering**) of Somaiya Vidyavihar University

Ms Kirti Mishra
Guide

23/01/25
Head of the Department

Dr Kavita Kelkar
Examiner

Date: 13-01-2025
Place: Mumbai-77

Abstract

In recent years, vision transformers have emerged as a powerful class of deep learning models for image classification tasks. Unlike traditional convolutional neural networks (CNNs), vision transformers rely on self-attention mechanisms to capture global dependencies within an image, making them highly effective for tasks such as object recognition and classification.

Two vision transformer models, namely ViT_B-16 and ViT_L-32, were meticulously trained and deployed for the classification of 400 different species of birds. Remarkably, the ViT_B-16 model demonstrated a testing accuracy of 95.603%, while its counterpart, the ViT_L-32 model, achieved an admirable 94.272% accuracy. The models were seamlessly integrated into a user-friendly web application using Streamlit, a popular framework for building interactive data applications. This deployment allows users, irrespective of their domain expertise, to effortlessly upload images and receive predictions regarding the depicted bird species.

Moreover, the application provides users with insights into the confidence level associated with each prediction, thereby enhancing the interpretability of the model's outputs. By offering a user-friendly interface coupled with cutting-edge vision transformer technology, the project not only facilitates bird species classification but also serves as a testament to the advancements in deep learning and its practical applications in real-world scenarios.

Keywords

Artificial Intelligence, Deep Learning, Vision Transformers, Image Classification, Bird Species Classification, Streamlit

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the implementation of a bird species classification project utilizing advanced deep learning techniques, specifically Vision Transformer (ViT) models. In an era where image recognition plays a pivotal role in various domains, accurately classifying bird species from images remains a challenging task due to the diverse visual characteristics and subtle distinctions among different species. Leveraging the power of ViT models, this project aims to develop a robust classification system capable of precisely identifying bird species based on visual cues extracted from images. The implementation encompasses a comprehensive pipeline, including data preprocessing, model training, evaluation, and deployment, with a focus on optimizing model performance and usability. Through this project, the efficacy of ViT models in addressing complex image recognition tasks and their potential applications in biodiversity monitoring and conservation efforts are explored.

1.1 Background

In the ever-evolving field of ornithology and biodiversity monitoring, the quest for efficient and accurate methods to identify and quantify various bird species has driven the development of innovative technologies. Traditional qualitative descriptions often prove inadequate in providing a comprehensive understanding of the avian world. In response to this need, the Wing Watch project emerges as a groundbreaking initiative that aims to leverage the power of Vision Transformers, an advanced class of deep learning models, to simplify bird species identification through the analysis of image features.

Ornithology, the scientific study of birds, has traditionally relied on manual methods for species identification, often hindered by subjectivity and limitations in accuracy. As biodiversity monitoring becomes increasingly vital, there is a pressing need to adopt technological advancements that can enhance the precision and efficiency of bird species identification. Traditional methods fall short in capturing the nuanced characteristics that define each species, leading to the emergence of a gap between qualitative descriptions and a comprehensive understanding of avian diversity.

The project addresses this gap by harnessing the capabilities of Vision Transformers (ViTs), a cutting-edge class of deep learning models. Vision Transformers have demonstrated exceptional prowess in image feature extraction and analysis, making them ideal candidates for

revolutionizing bird species identification. By adopting these advanced technologies, Wing Watch seeks to create a robust system that not only quantifies the qualitative attributes of different bird species but also establishes a foundation for accurate and reliable species identification.

1.2 Problem Statement

The problem statement of bird species image classification revolves around the challenge of accurately discerning and categorizing various avian species depicted in images. This task entails deciphering intricate visual cues, such as plumage patterns, beak shapes, wing morphology, and coloration variations, among others, to differentiate between hundreds or even thousands of distinct bird species. Given the vast diversity and nuanced features present across avian taxa, coupled with the potential for environmental factors like lighting conditions and background clutter to confound image interpretation, the task of automated bird species classification poses formidable computational and algorithmic challenges.

Moreover, the importance of precise bird species identification extends beyond mere academic or recreational interests, encompassing critical applications in ecological research, biodiversity monitoring, conservation efforts, and wildlife management. Accurate classification of bird species from images serves as a foundational tool for understanding avian biodiversity, tracking population dynamics, assessing habitat health, identifying species distributions, and evaluating the impacts of environmental disturbances or anthropogenic activities on avifauna.

Furthermore, the task of bird species image classification is inherently interdisciplinary, drawing upon insights and methodologies from fields such as computer vision, machine learning, ornithology, and ecology. Integrating advanced deep learning techniques, like convolutional neural networks (CNNs) or Vision Transformers (ViTs), with domain-specific knowledge of avian taxonomy and morphology holds promise for developing highly accurate and efficient classification systems capable of handling large-scale image datasets encompassing diverse bird species.

In summary, the problem statement of bird species image classification encapsulates the multifaceted challenge of accurately and efficiently categorizing avian taxa depicted in images, with implications spanning ecological research, conservation science, and biodiversity monitoring. Addressing this challenge requires the integration of cutting-edge computational

methods, domain expertise, and interdisciplinary collaboration to develop robust and scalable solutions capable of advancing our understanding and stewardship of avian biodiversity.

1.3 Scope

The scope of the project encompasses the development of an advanced image classification system applicable to a wide range of domains and applications. The project aims to address the complex task of accurately identifying and categorizing objects depicted in images, leveraging state-of-the-art deep learning techniques. Key components within the scope include dataset acquisition and preprocessing, model selection and architecture design, training and evaluation procedures, optimization and fine-tuning strategies, deployment and integration into user-friendly interfaces, and comprehensive documentation and knowledge transfer. By adopting a holistic approach to image classification, the project seeks to develop robust and scalable solutions capable of advancing various fields, including computer vision, remote sensing, medical imaging, environmental monitoring, and more. The project's overarching goal is to contribute to the advancement of image understanding technologies and facilitate their widespread adoption in diverse real-world applications, ultimately fostering innovation and enhancing decision-making processes across multiple domains.

1.4 Objectives

The objectives of the project are delineated as follows:

i. **Comprehensive Augmented Database:** The primary objective is to construct an extensive augmented database encompassing standard image features for approximately 400 bird species. This database serves as the foundational resource for training the Vision Transformer models. By ensuring a diverse and representative set of features, the augmented database facilitates accurate species identification, enhancing the robustness and generalization capabilities of the classification system.

ii. **Training Vision Transformer Models:** The project aims to optimize and implement an efficient classifier layer that maximizes the utilization of pretrained Vision Transformer models. By leveraging the full capabilities of the models, including learned features and hierarchical representations, the optimized classifier layer enhances the accuracy and discriminative power of species classification. This optimization process ensures that the models can effectively translate extracted features into precise species predictions, further improving the overall performance of the system.

iii. **User-Friendly Web Application:** The project seeks to democratize access to its advanced bird species identification system through the development of a user-friendly web application. Utilizing Streamlit, the interface will offer an intuitive platform for users to interact with the classification system effortlessly. By enabling users to upload bird images, select desired Vision Transformer models, and receive prompt and visually appealing results, the web application enhances accessibility and usability, catering to a diverse range of stakeholders, including researchers, conservationists, and citizen scientists.

iv. **Confidence Score and Result Display:** In addition to predicting the bird species based on uploaded images, the system will provide users with a confidence score, indicating the model's certainty regarding the prediction. This feature enhances transparency and allows users to gauge the reliability of the classification results. By providing insight into the model's confidence level, users can make informed decisions based on the classification outcomes, fostering trust and confidence in the system's capabilities.

Overall, the objectives of the project aim to establish a robust and user-friendly bird species identification system empowered by advanced deep learning techniques. Through the creation of a comprehensive database, optimization of the classifier layer, development of a user-friendly web application, and provision of confidence scores, the project seeks to enhance the accuracy, accessibility, and transparency of bird species classification, contributing to advancements in avian biodiversity research and conservation efforts.

1.5 Hardware and Software Requirements for Development:

The hardware requirements are as follows:

i. **Processor (CPU):** A multi-core processor (quad-core or higher) is essential for efficient computation during model training and inference, as deep learning tasks are computationally intensive.

ii. **RAM:** To handle large datasets and deep learning model operations effectively, a minimum of 16GB RAM is recommended. Higher RAM capacity can further enhance performance, especially when working with larger models or datasets.

iii. **GPU (Graphics Processing Unit):** A dedicated GPU is highly recommended for accelerating deep learning tasks, as it significantly speeds up computations compared to CPU-only processing. NVIDIA GPUs, such as the GeForce or Tesla series, are commonly used for TensorFlow and Keras operations due to their excellent support for parallel processing.

iv. **Storage:** Adequate storage space is required for storing datasets, model checkpoints, and other project-related files. Solid-state drives (SSDs) are preferable over traditional hard disk drives (HDDs) for faster data access, which can expedite training and inference processes.

v. **Internet Connection:** A stable and high-speed internet connection is necessary for accessing cloud-based services, downloading datasets, and staying updated with the latest software libraries and frameworks.

The software requirements are as follows:

i. **Operating System:** The project can be developed on various operating systems, including Windows, Linux, or macOS, based on user preference and compatibility with the required software.

ii. **Python:** Install the latest version of Python (3.7 or above), as it serves as the primary programming language for the entire project, offering a wide range of libraries and tools for deep learning and data processing tasks.

iii. **Frontend - Streamlit:** Install Streamlit using pip install streamlit. Streamlit simplifies the development of interactive web applications, providing an intuitive interface for users to interact with the classification system.

iv. **Backend - TensorFlow and Keras:** Install TensorFlow with GPU support using pip install tensorflow. TensorFlow is a popular deep learning framework that provides comprehensive support for building and training neural networks. TensorFlow includes Keras as its high-level neural networks API, making it easy to build and deploy deep learning models without the need for separate installation.

v. **Notebook Environment - Google Colab or Kaggle:** Access to Google Colab or Kaggle, a cloud-based notebook environment, is beneficial for development and training purposes, as it provides free GPU and TPU support, enabling faster model training and experimentation.

vi. **Image Processing and Computer Vision:** Install the Python Imaging Library using pip install Pillow for handling various image file formats and performing basic image processing tasks. Install OpenCV for advanced image processing and computer vision tasks, including object detection and feature extraction.

vii. **Additional Python Libraries:** Install NumPy for efficient numerical operations and array manipulation. Install Matplotlib for creating visualizations, such as plots and charts, to analyze model performance and results.

1.6 Methodology:

To figure out how confident your model is in its prediction, we can look at the activation values from the last layer (also known as logits). These values represent how strongly the model believes in each class, but they need to be turned into probabilities to make sense to us.

Here's how we can do that:

1. Start with the activation values (logits):
   After your model makes a prediction, you'll get a set of numbers (the activation values) for each possible class. These are raw scores, but they don't quite tell you the probability just yet.

2. Use the softmax function:
   To turn those raw scores into probabilities (values between 0 and 1), we apply something called the "softmax" function. It basically adjusts the scores so they sum up to 1 and shows us how likely the model thinks each class is.

   Here's how it works:
   - For each class k, we take the exponent of the activation value zk, which helps amplify larger values.
   - Then, we divide that by the sum of all the exponentiated scores (so the probabilities add up to 1).

   Mathematically, it looks like this:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{C} e^{z_j}}$$

   This gives us the probability for each class k, where C is the total number of classes.

3. Confidence level for the predicted class:
   The class with the highest probability is the one the model is most confident in. So, we just pick the class that has the highest score after applying softmax, and that's our predicted class.

   Finally, the confidence level is simply the probability of that predicted class:

$$\text{Confidence Level} = p_{\hat{y}} = \frac{e^{z_{\hat{y}}}}{\sum_{j=1}^{C} e^{z_j}}$$

   The higher this number, the more confident the model is about its prediction.

Chapter 2

Literature Survey

This chapter presents the literature survey conducted as part of the project on bird species image classification. Bird species classification from images represents a crucial task with diverse applications in ecology, conservation biology, and computer vision. The ability to accurately identify bird species from images is essential for biodiversity monitoring, habitat assessment, and population studies. However, despite significant advancements in deep learning and computer vision techniques, challenges such as intra-species variability, background clutter, and occlusions persist, necessitating innovative approaches and robust methodologies. This literature survey aims to explore existing research efforts, methodologies, and advancements in bird species image classification. By reviewing a diverse range of scholarly articles, research papers, and technical reports, the survey seeks to identify key trends, methodologies, and best practices in the field. Furthermore, the literature review aims to elucidate gaps in knowledge, unresolved challenges, and emerging research directions, providing valuable insights for the development of novel approaches and methodologies in bird species classification.

Paper	Objective	Dataset	Model/Algorithm	Result/Conclusion
1) Stacked Res2Net-CBAM with Grouped Channel Attention for Multi-label Bird Species Classification	The objective of the study is to enhance the classification performance of multi-label bird species using a novel framework that combines Stacked Res2Net, Convolutional block attention module (CBAM), and Grouped Channel Attention (GCA). The aim is to address the challenges of detecting	1) The audio recordings for the study were collected from the bird sound database of the Xeno-Canto Foundation 2. 2) The dataset consists of various bird species with well-defined call structures, covering chirps, whistles, blocks, warbles, and clicks. 3) The dataset was pre-processed by normalizing all files to a sample rate of 16 kHz and converting them to mono	1) The proposed framework utilizes Stacked Res2Net-CBAM with Grouped Channel Attention (GCA) using a sliding window analysis approach. 2) The model incorporates Res2Net, CBAM, and GCA to improve the detection performance of multiple overlapping bird species in audio recordings. 3) The system architecture includes a stack	1) The proposed Stacked Res2Net-CBAM with GCA framework significantly improved the classification performance of multi-label bird species. 2) The model demonstrated enhanced capabilities in handling variations in audio recordings, such as background noise and overlapping calls. 3) The results indicated that the proposed approach outperformed existing models in

	multiple overlapping bird species in audio recordings with variations in background noise and the presence of other calls 4.	with 32-bit resolution 2.	with N=1 and 64 filters, a reduction ratio of 1, and groups=4.	detecting multiple bird species in challenging audio environments 4.
2) Bird Species Classification Based on Color Features	Develop a method for classifying bird species based solely on color features extracted from unconstrained images.	UB-200-2011: This dataset contains images of 200 bird species from North America, with annotations for bounding boxes, part locations, and attributes like color and pose. It is a popular benchmark dataset for bird species classification tasks. FGVC Birds: This dataset includes images of 342 bird species with bounding boxes and class labels. It offers a larger variety of species compared to CUB-200-2011 but might have lower image quality.	Two approaches are explored: Concatenation: Combines features from all channels into a single feature vector for classification. Individual Classifiers: Trains separate SVMs for each color channel and combines their outputs.	Results: •ResNet152V2: Achieves the highest accuracy (95.45%) but suffers from a higher loss (0.8835). •DenseNet201: Exhibits slightly lower accuracy (95.05%) but has a lower loss (0.6854), indicating potentially better generalization. Conclusion: • DenseNet201 is chosen as the most suitable model for real-life bird image classification due to its balanced accuracy and loss. • The system aims to be developed further as a web application to aid bird photographers in identifying captured species.

3) Bird Image Classification using Convolutional Neural Network Transfer Learning Architectures	Utilizes pre-trained CNN models (ResNet152V2, InceptionV3, DenseNet201, MobileNetV2) instead of training from scratch, leveraging their existing knowledge.	Uses a dataset containing 58,388 images belonging to 400 bird species.	1) Adds an output layer to the chosen pre-trained models for bird species classification. 2) Trains the models using the provided dataset.	1) Compares the performance of all four models based on accuracy and loss. 2) ResNet152V2: Achieves the highest accuracy (95.55%) but suffers from a higher loss (0.905). 3) DenseNet201: Exhibits slightly lower accuracy (95.05%) but has a lower loss (0.6854), indicating potentially better generalization.
4) Bird Species Classification Using Deep Learning Approach		The dataset consisted of images of 20 different bird species obtained from online sources. A total of 7637 images were used for training and 1853 images were used for testing.	The model used a Convolutional Neural Network (CNN) architecture. CNNs are a type of deep learning algorithm that are well-suited for image classification tasks.	The model achieved an accuracy of 98.75% on the test dataset. The authors concluded that the CNN-based model is an effective approach for bird species classification and has the potential to be used in applications such as bird identification for tourists and conservation efforts.
5) A comparative study on deep learning techniques for bird species recognition.	Experimented with five different models: MobileNet, AlexNet, InceptionResNet V2,	The experiment used a bird species dataset from Kaggle containing 11,788 images belonging to 200 classes. Due	A comparative study on the performance of various deep learning models for bird species recognition using images.	EfficientNet outperformed other models with a test accuracy of 87.13%. MobileNet and EfficientNet were

	Inception V3, and EfficientNet.	to data insufficiency, data augmentation was performed to increase the number of images to 40,000.	The models were trained and tested on a split of the dataset (70% training, 20% validation, and 10% testing).Exploring different deep learning models and hyperparameter tuning. Including metrics like precision, recall, and F1-score for a more comprehensive evaluation. They also suggest investigating the performance of these models with an increasing number of bird classes.	the fastest training models, taking 4 minutes per epoch compared to 10 minutes for the other models.

Table 2.1: Literature survey

Chapter 3

Project Details

This chapter presents the implementation details and system architecture of a bird species classification project utilizing Vision Transformer (ViT) models. The primary objective of the project was to develop a robust classification system capable of accurately identifying bird species from images. The system architecture comprised several interconnected components, including data preprocessing, model training, evaluation, and deployment.

3.1 : Proposed System Model/Architecture

System Overview:

The system overview of this project provides a holistic understanding of its architecture, components, and functionalities. At its core, the project is designed to facilitate the accurate classification of bird species from images through the integration of advanced deep learning techniques and user-friendly interfaces. The system comprises several interconnected modules, each contributing to the overall functionality and effectiveness of the bird species classification system.

i. **Data Acquisition and Preprocessing:** The system begins with the acquisition and preprocessing of bird image datasets. This involves sourcing or curating a comprehensive dataset containing images of various bird species. The images are then preprocessed to standardize formats, sizes, and orientations, ensuring uniformity and compatibility for model training and inference.

ii. **Model Development and Training:** The heart of the system lies in the development and training of deep learning models for bird species classification. Vision Transformer (ViT) models are utilized for their efficacy in handling large-scale image datasets and capturing intricate visual patterns. The models are trained on the preprocessed image datasets, leveraging transfer learning and optimization techniques to achieve high accuracy and robustness.

iii. **User Interface Development:** The system features a user-friendly web application interface developed using Streamlit. This interface allows users to interact with the classification system seamlessly, enabling them to upload bird images, select desired ViT models, and receive real-time classification results with associated confidence scores.

The interface is designed to be intuitive and accessible to users with varying levels of technical expertise.

iv. **Result Presentation and Interpretation:** Upon uploading an image, the system processes it through the trained ViT models and generates classification results, including predicted bird species and confidence scores. These results are presented to the user in a visually appealing format, allowing for easy interpretation and analysis. Additionally, users have the option to view detailed classification metrics and insights into model performance.

v. **Scalability and Deployment:** The system is designed to be scalable and deployable in various environments, including cloud-based platforms or on-premises servers. Deployment considerations ensure that the system can accommodate growing user demands and effectively handle large volumes of image data while maintaining high performance and reliability.

Overall, the system overview encapsulates the intricate interplay between data processing, model development, user interface design, and deployment considerations, culminating in a comprehensive and user-centric bird species classification system powered by advanced deep learning techniques.

3.2 Software Project Management Plan

i. Project Overview:

- Objective: Develop an efficient system for detecting bird classification.
- Scope: Extract features, train machine learning models, and evaluate the model's performance.

ii. Project Team:

- Project Manager: Vedant Chavan
- Data Scientist: Atharv Chandane, Kunal Chaturvedi
- Developer: Sakshi Borade
- Quality Assurance: Sahil Chauhan

iii. Timeline and Milestones:

- Phase 1 - Feature Extraction: 15 January 2024 to 25 January 2024
- Phase 2 - Model Training: 26 January 2024 to 5 February 2024
- Phase 3 - Evaluation: 5 February 2024 to 16 February 2024

iv. Resource Allocation:

- Project Manager: Oversight, coordination, and communication.

- Data Scientist: Feature extraction and model development.
- Developer: Implementation of system functionalities.
- Quality Assurance: Testing and validation.

v. **Risk Management:** The identified risks are as follows:

- Data inconsistency.
- Model overfitting.
- Resource constraints.

vi. **Mitigation Strategies:**

- Data preprocessing for consistency.
- Regularization techniques for model optimization.
- Prioritization and efficient use of resources.

vii. **Tools and Technologies:**

- Python, TensorFlow, Keras
- Pandas and NumPy for data manipulation.
- Scikit-learn for machine learning.
- Matplotlib and Seaborn for visualization.

viii. **Development and Deployment Phases:**

- Phase 1 - Feature Extraction:
  Implement feature extraction mechanisms.
  Verify data consistency.
- Phase 2 - Model Training:
  Choose and implement a machine learning model.
  Train the model using the dataset.
- Phase 3 - Evaluation:
  Evaluate the model's performance.

3.3 Software Design Document

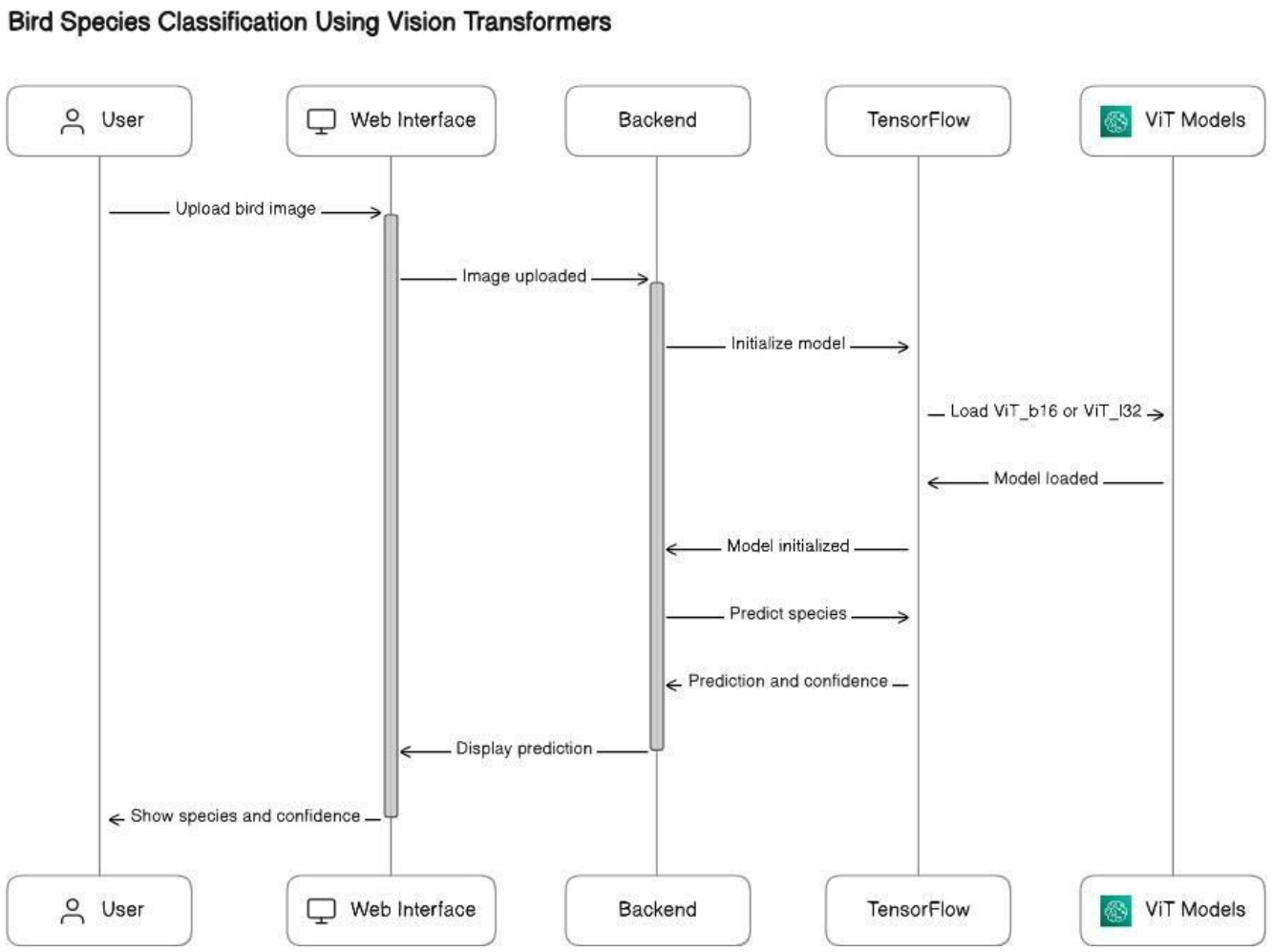

Fig 3.1: System Flow Diagram

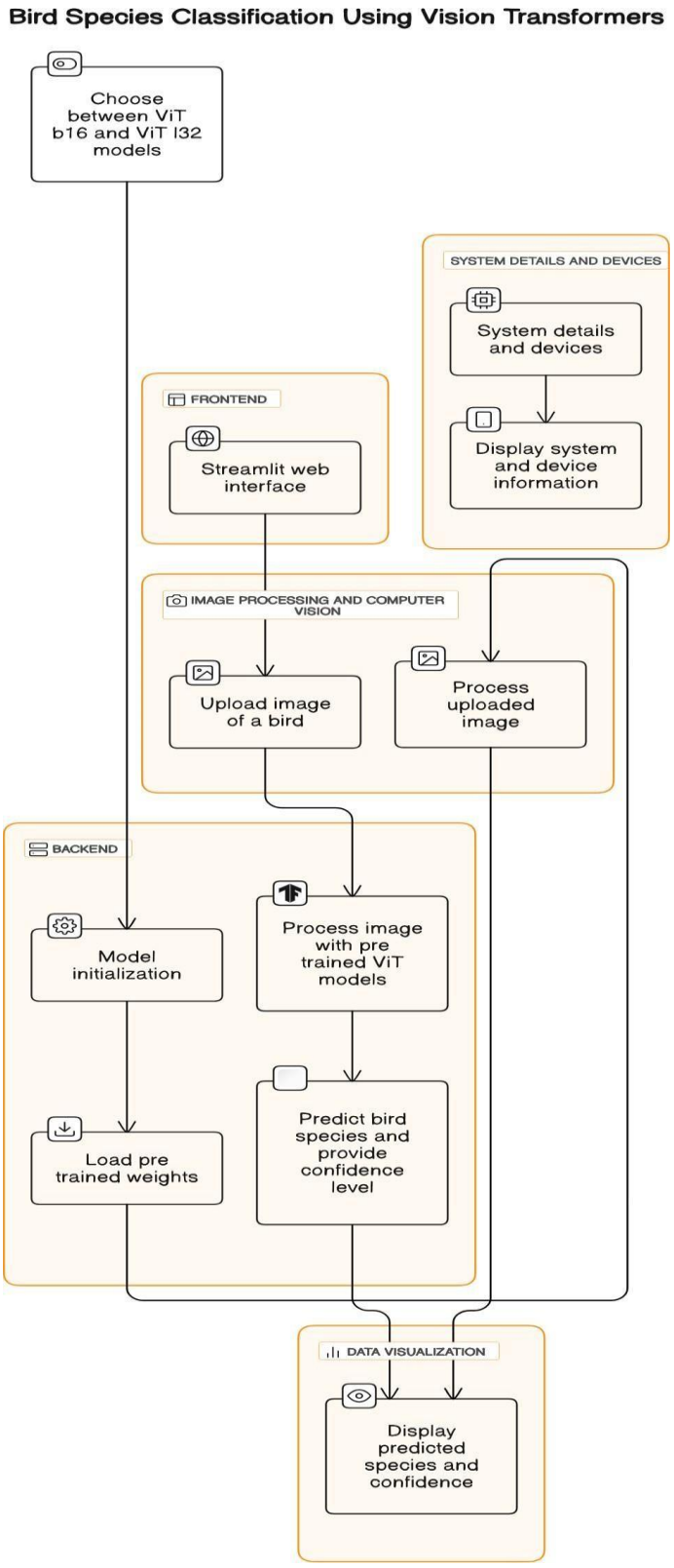

Fig 3.2: System Functionality Workflow

Chapter 4

Implementation and Experimentation

This chapter presents the implementation and experimentation process conducted to enhance the performance of Vision Transformer (ViT) models for the classification of bird species. The utilization of augmentations and rescaling techniques stands as a pivotal aspect in refining the robustness and generalization capabilities of these models. By systematically integrating augmentations and rescaling methods into the training pipeline, we aimed to bolster the models' ability to accurately classify diverse bird species depicted in varying environmental conditions and image compositions.

Through meticulous experimentation and evaluation, we scrutinized the impact of augmentations and rescaling on the performance metrics of ViT_B-16 and ViT_L-32 models. Specifically, we measured the effects on key metrics such as Recall Score, Precision Score, and F1 Score, providing insights into the efficacy of these techniques in improving classification accuracy and model robustness.

Furthermore, this chapter elucidates the rationale behind the selection of specific augmentation techniques and rescaling strategies, outlining their theoretical underpinnings and practical implications. By systematically detailing the implementation methodology and experimental setup, we offer a comprehensive understanding of the augmentation and rescaling techniques employed to refine the performance of ViT models in bird species classification tasks.

In addition to model refinement, this chapter also delves into the deployment aspect, where the trained ViT models are seamlessly integrated into a user-friendly web application using Streamlit, a popular framework for building interactive data applications. Through this deployment, users can upload images and receive real-time predictions regarding the depicted bird species, along with confidence levels associated with each prediction. This integration not only facilitates easy access to the classification capabilities of ViT models but also showcases their practical applicability in the domain of ornithology and beyond.

4.1 : Proposed System Model Implementation

Libraries Used:

- **numpy (np):** NumPy is a fundamental package for numerical computing in Python. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.
- **pandas (pd):** Pandas is a versatile and powerful data analysis and manipulation library for Python. It offers data structures like DataFrame, which allows for easy handling and

manipulation of structured data, making it indispensable for data preprocessing and analysis tasks.

- **os:** The os module in Python provides a way to interact with the operating system. It allows for tasks such as file and directory manipulation, environment variable access, and execution of system commands, making it essential for managing files and directories in a Python program.
- **random:** The random module in Python provides functions for generating pseudo-random numbers, selecting random elements, shuffling sequences, and more. It is commonly used in simulations, games, and cryptography to introduce randomness into programs.
- **dataclasses:** The dataclasses module, introduced in Python 3.7, simplifies the creation of classes for storing data by automatically generating special methods such as init(), repr(), and eq(). It enhances code readability and reduces boilerplate code when defining simple data structures.
- **matplotlib.pyplot (plt):** Matplotlib is a comprehensive plotting library for Python that produces high-quality static, animated, and interactive visualizations. The pyplot submodule provides a MATLAB-like interface for creating plots and charts, making it widely used for data visualization tasks.
- **seaborn (sns):** Seaborn is a statistical data visualization library built on top of matplotlib. It provides a high-level interface for creating informative and visually appealing statistical graphics, making it popular among data scientists and analysts for exploratory data analysis.
- **cv2:** OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision tasks. It offers a wide range of tools and algorithms for image processing, object detection, video analysis, and more.
- **tensorflow (tf):** TensorFlow is an open-source deep learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models, including neural networks, convolutional networks, and recurrent networks.
- **keras:** Keras is a high-level neural networks API written in Python and capable of running on top of TensorFlow, among other backends. TensorFlow's implementation of Keras offers a user-friendly interface for building and training deep learning models, making it accessible to beginners and experts alike.
- **vit_keras:** The vit_keras library provides tools and utilities for working with Vision Transformer (ViT) models in Keras. It includes implementations of ViT models, as well

as utilities for training, visualization, and understanding attention mechanisms in these models.

- **sklearn:** Scikit-learn is a popular machine learning library for Python that provides simple and efficient tools for data mining and data analysis. It offers a wide range of algorithms for classification, regression, clustering, dimensionality reduction, and more, along with utilities for model evaluation and validation.
- **warnings:** The warnings module provides functions to control warning messages in Python. It allows developers to filter, ignore, or handle warning messages during program execution, helping to identify potential issues or unexpected behavior in code.
- **streamlit (st):** Streamlit is a Python library for building interactive web applications with simple Python scripts. It enables developers to create and deploy data-driven applications quickly and easily, without needing to write HTML, CSS, or JavaScript code.
- **platform:** The platform module provides information about the underlying platform on which Python is running. It allows developers to access details such as the operating system, architecture, and hardware, which can be useful for platform-specific optimizations or compatibility checks.
- **pickle:** The pickle module in Python is used for serializing and deserializing Python objects. It allows objects to be saved to a file and loaded back into memory, preserving their state across different sessions or environments.

Dataset Used:

- **Reduction of Dataset Size:** Due to limitations in GPU memory capacity, it was necessary to reduce the size of the dataset to ensure feasibility for training neural network models. The dataset was reduced to a subset containing 71,407 images, covering 400 bird species.
- **Distribution of Images:** The reduced dataset was partitioned into three subsets: training, validation, and testing, maintaining a standard split ratio. Specifically, 56,987 images were allocated to the training set, 7,210 images to the validation set, and 7,210 images to the testing set. This partitioning ensured adequate representation of each bird species across the different subsets, maintaining the integrity of the dataset.
- **Train-Validation-Test Ratio:** The train-validation-test ratio was set to 80:10:10, adhering to best practices in machine learning model development. This ratio ensures that the models are trained on a sufficiently large dataset while also providing separate subsets for validation and testing to assess performance and generalization.

Exploratory Data Analysis (EDA):

Exploratory data analysis (EDA) constitutes a foundational phase in the lifecycle of any data-driven project, and its role in the classification of bird species is pivotal. It involves a meticulous and thorough examination of the dataset, employing various statistical and visualization techniques to unravel intricate patterns, distributions, and underlying structures inherent within the data. Within the context of classifying bird species, EDA assumes heightened significance owing to several compelling reasons.

A fundamental aspect of EDA involves comprehending the distribution of images across the myriad bird species represented in the dataset. By discerning the classes housing a higher abundance of images, researchers can glean crucial insights into the dataset's composition and ensure an equitable representation of different bird species. This equitable distribution is paramount to forestall biases during subsequent model training phases, fostering a more robust and unbiased learning process that can adeptly generalize to unseen data.

Moreover, a granular exploration of classes with the highest image counts permits a nuanced understanding of image quality and variability inherent within each category. This exploration facilitates the detection of subtle nuances, such as variations in lighting conditions, diverse bird poses, or complex backgrounds. By identifying these inherent challenges, researchers can meticulously devise preprocessing strategies tailored to address such intricacies, thereby fortifying the model's resilience and bolstering its predictive performance.

Furthermore, EDA serves as an invaluable mechanism for unearthing outliers or anomalies lurking within the dataset. By meticulously scrutinizing the data landscape, researchers can pinpoint and remediate aberrations that might otherwise compromise the dataset's integrity. The curation of a clean and representative dataset, devoid of outliers, is instrumental in ensuring a robust training process that fosters the development of a highly accurate and reliable classification model.

The classes with the most number of images is then examined:

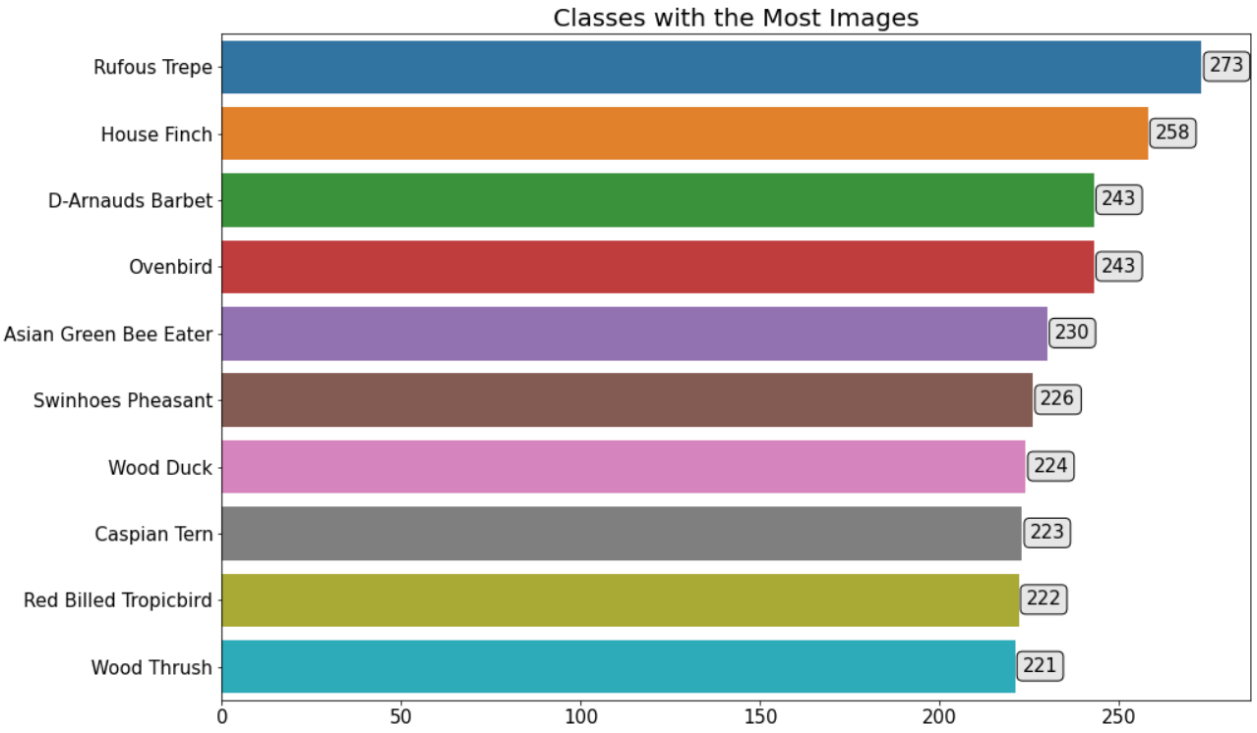

Fig 4.1: Bird Species with the Most Number of Images in the Original Dataset

Dataset Preprocessing:

- **Shuffling and Randomization:** To prevent any bias or order effects in the dataset, the images were shuffled randomly before partitioning into training, validation, and testing subsets. Randomization ensures that each subset contains a diverse representation of bird species and image characteristics, reducing the risk of overfitting during model training.
- **Normalization and Standardization:** Prior to training the neural network models, the pixel values of the images were normalized and standardized. Normalization scales the pixel values to a range between 0 and 1, facilitating convergence during model training. Standardization ensures that the pixel values have a mean of 0 and a standard deviation of 1, further stabilizing the training process and improving model performance.
- **Data Augmentation:** To increase the diversity and variability of the training data, data augmentation techniques were applied to the images in the training set. Augmentation techniques such as rotation, flipping, cropping, and zooming were used to generate additional training samples, thereby enhancing the model's ability to generalize to unseen data.
- **Data Loading and Batch Generation:** Finally, the preprocessed dataset was loaded into memory and organized into batches for efficient training of the neural network models. Batch generation allows the models to process a subset of the dataset at a time, reducing memory consumption and accelerating the training process.

After the dataset split the images in the training, validation and testing set can be visualized:

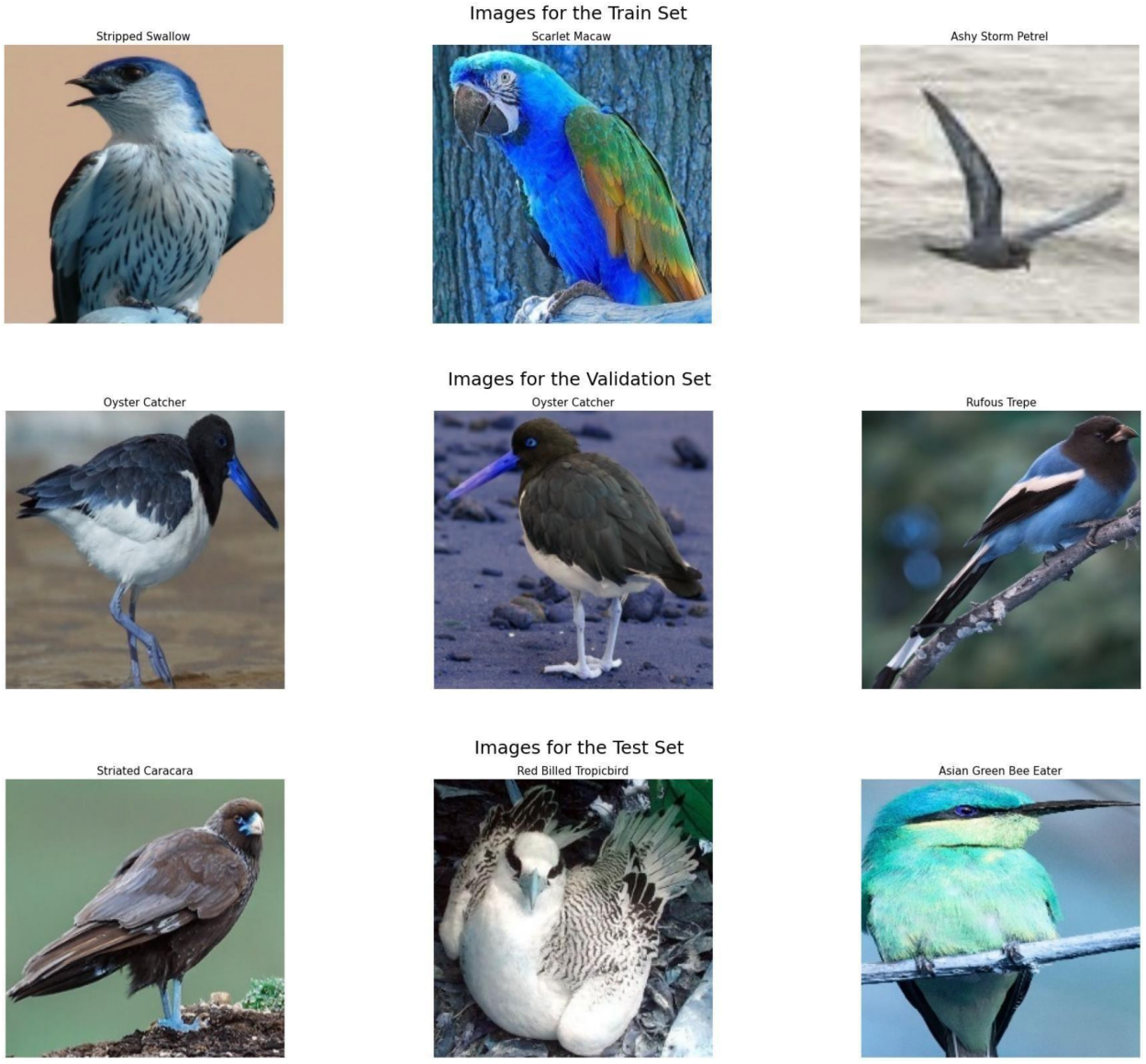

Fig 4.2: Images in the training, validation and testing splits

Training the Vision Transformer Models:

Vision Transformers (ViTs) are a class of deep learning models that have gained significant traction in the field of computer vision, offering a novel approach to image understanding and processing. Unlike traditional Convolutional Neural Networks (CNNs), which have long been the cornerstone of image recognition tasks, ViTs eschew the hierarchical convolutional layers in favor of self-attention mechanisms inspired by Transformer models originally developed for natural language processing tasks.

At the heart of a Vision Transformer lies the Transformer architecture, which was first introduced in the context of natural language processing (NLP). The Transformer architecture fundamentally consists of two key components: the self-attention mechanism and the feed-forward neural network. In NLP tasks, self-attention enables the model to capture long-range dependencies between different words in a sequence, facilitating robust contextual understanding.

When applied to computer vision tasks, such as image classification, ViTs leverage self-attention mechanisms to capture global dependencies within an image, allowing the model to attend to relevant regions and features regardless of their spatial proximity. This stands in contrast to CNNs, which rely on local receptive fields and hierarchical feature extraction through convolutional layers.

The architecture of a Vision Transformer typically consists of a stack of transformer encoder blocks. Each transformer encoder block comprises multiple layers of self-attention modules followed by position-wise feed-forward neural networks. The self-attention mechanism enables the model to aggregate information from all parts of the image, capturing both local and global context, while the feed-forward neural networks process the aggregated features to generate meaningful representations.

One notable characteristic of Vision Transformers is their ability to handle images of arbitrary size through the use of fixed-size patches. These patches serve as the input tokens for the ViT model, allowing it to process images in a sequence-like manner. By dividing the input image into patches and flattening them into sequences, ViTs can effectively leverage the self-attention mechanism to capture spatial relationships between patches.

Training a Vision Transformer typically involves pretraining on large-scale datasets, followed by fine-tuning on task-specific data. During pretraining, ViTs are exposed to a diverse range of images, enabling them to learn rich visual representations that can generalize well across different tasks. Fine-tuning involves adapting the pretrained ViT to the target task, such as image classification, object detection, or segmentation, by updating the model parameters on task-specific data.

Overall, Vision Transformers represent a paradigm shift in computer vision, offering a flexible and powerful framework for image understanding and processing. Their ability to capture global dependencies and handle images of arbitrary size makes them well-suited for a wide range of tasks, from image classification to image generation, with promising results across various benchmarks and datasets.

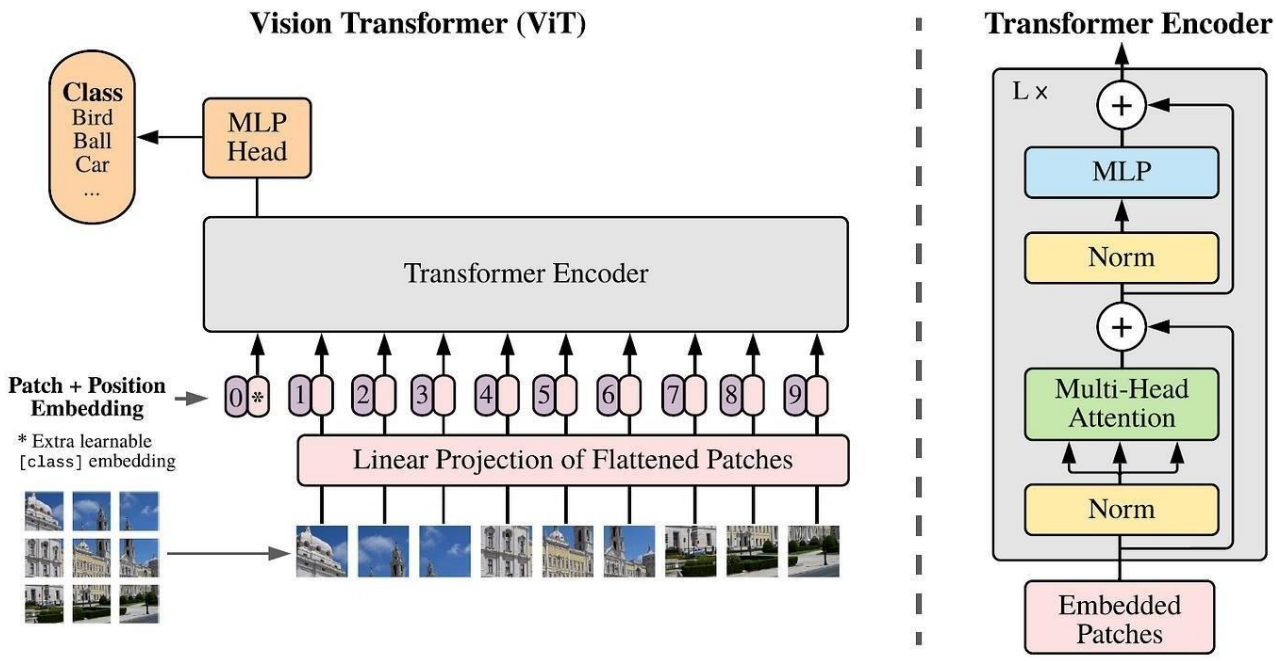

Fig 4.3: Architecture of a Vision Transformer (ViT)

For the purpose of bird classification, 2 vision transformer models are trained:

i. **ViT_B-16:** ViT_B-16, short for Vision Transformer Base-16, is a smaller and more computationally efficient version of the Vision Transformer architecture. It comprises a relatively shallow stack of transformer encoder blocks with 12 layers, making it lighter in terms of parameters and computations compared to larger variants. Despite its reduced complexity, ViT_B-16 retains the essential self-attention mechanisms and feed-forward neural networks characteristic of Vision Transformers, enabling it to capture global dependencies within images and generate meaningful representations. This makes ViT_B-16 suitable for scenarios where computational resources are limited or where a more lightweight model is preferred, without sacrificing performance significantly.

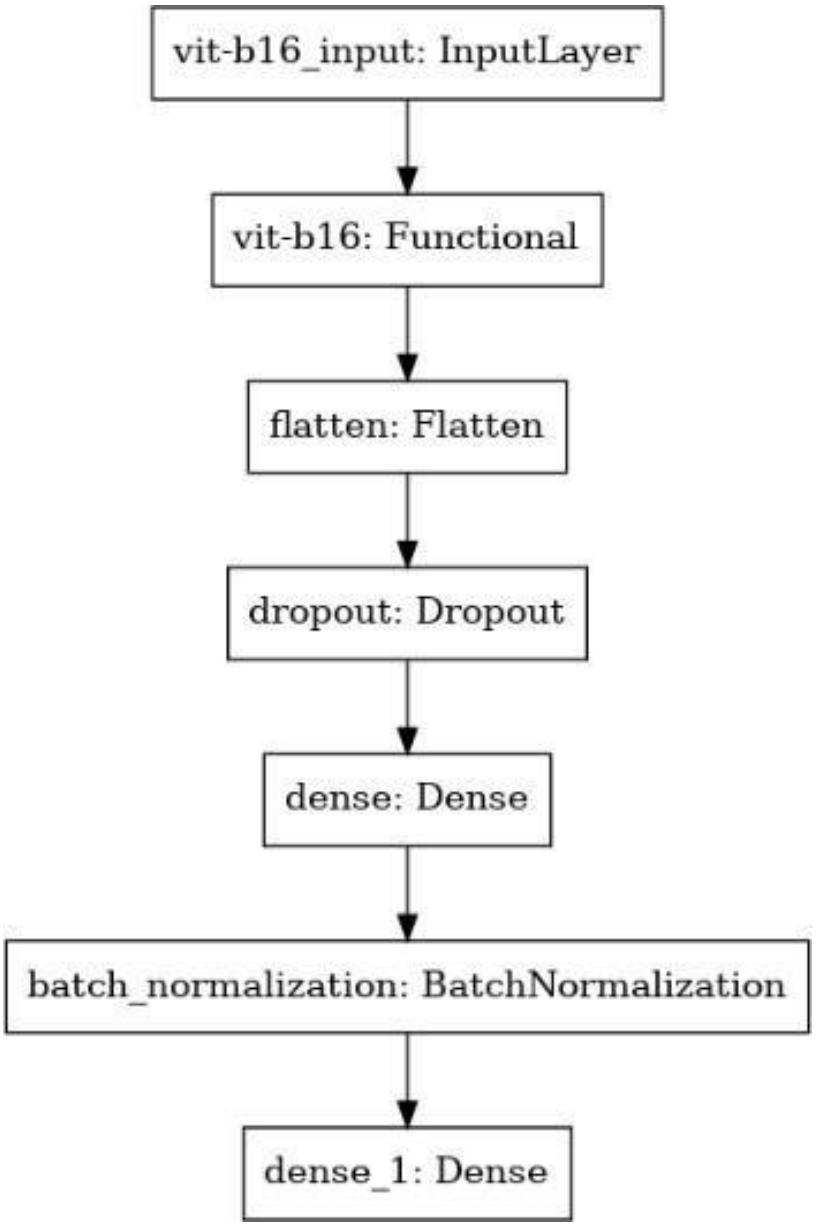

Fig 4.4: Neural Network Layers with ViT_B-16 as a backbone

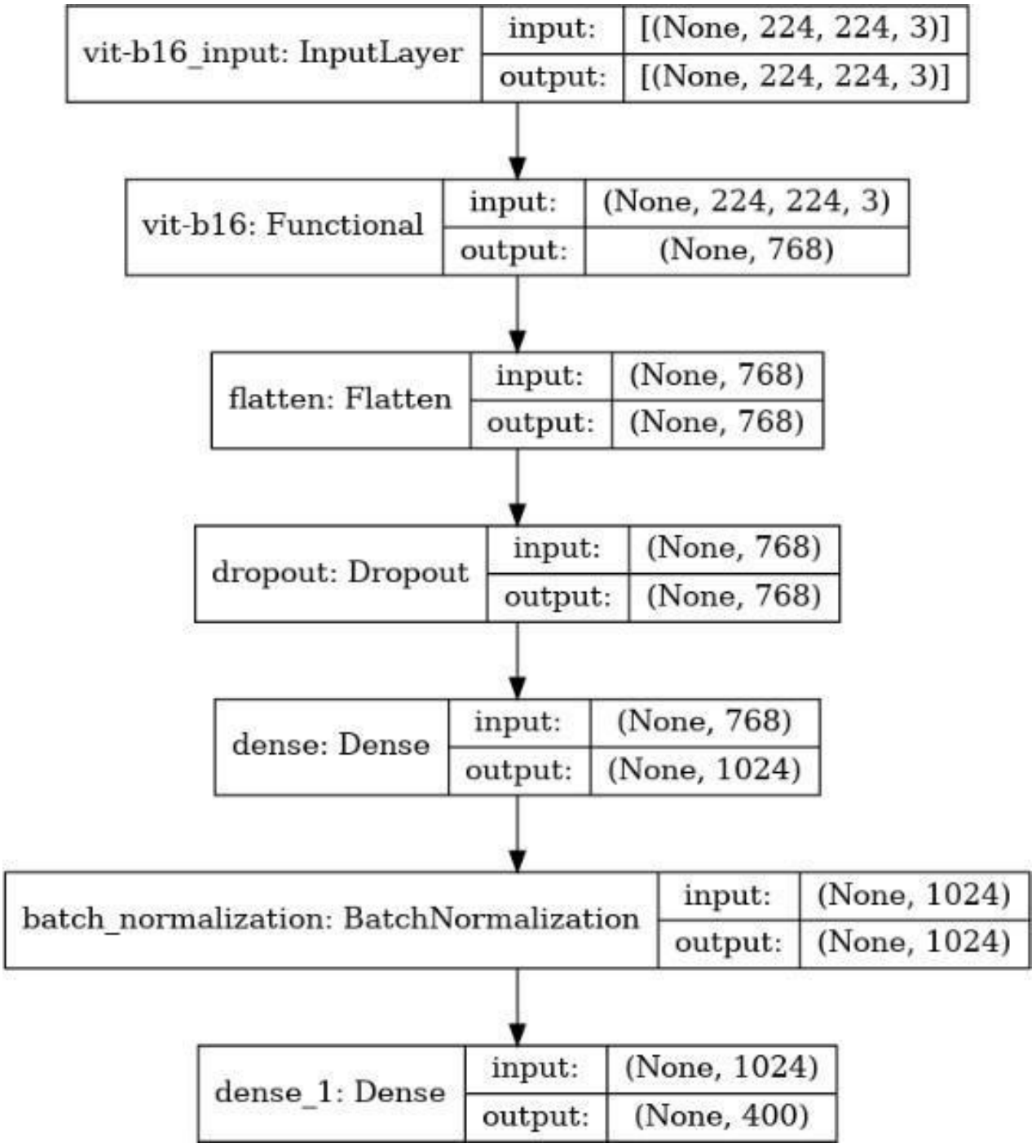

Fig 4.5: ViT_B-16 Neural Network Layers with input and output shapes

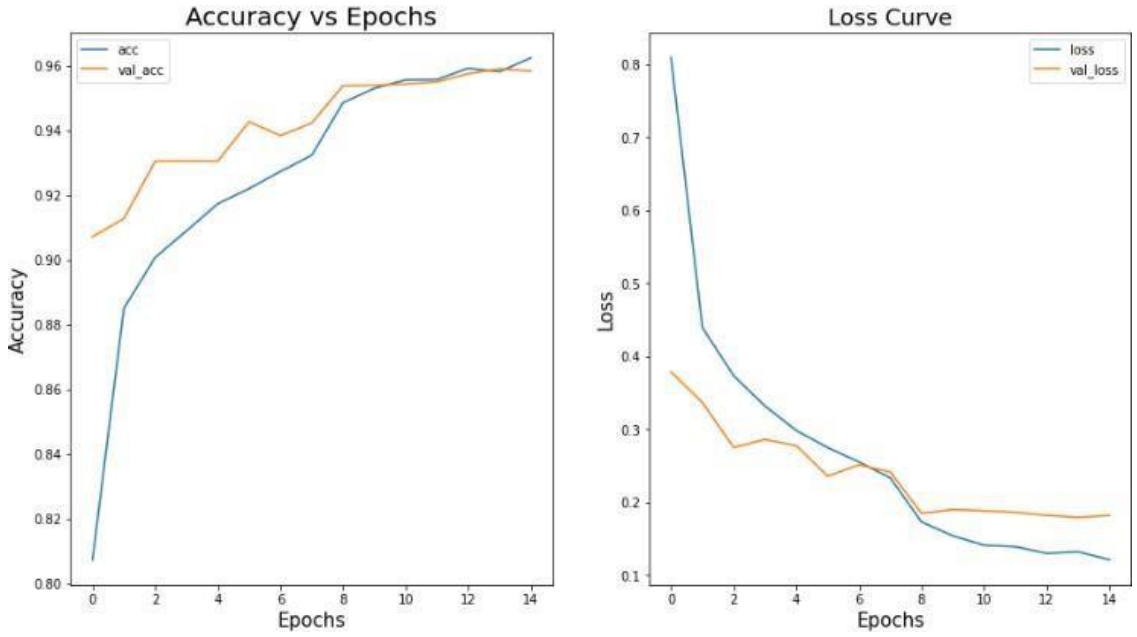

Fig 4.6: Accuracy and Loss Curves with ViT_B-16

ii. **ViT_L-32:** On the other hand, ViT_L-32, short for Vision Transformer Large-32, represents a larger and more expressive variant of the Vision Transformer architecture. It comprises a deeper stack of transformer encoder blocks with 32 layers, resulting in a higher model capacity and greater representational power. With its increased depth and complexity, ViT_L-32 can capture more intricate patterns and finer details within images, making it well-suited for tasks that demand higher levels of abstraction and semantic understanding. While ViT_L-32 may require more computational resources and training time compared to smaller variants, its superior performance and ability to handle complex datasets justify its usage in applications where accuracy and robustness are paramount.

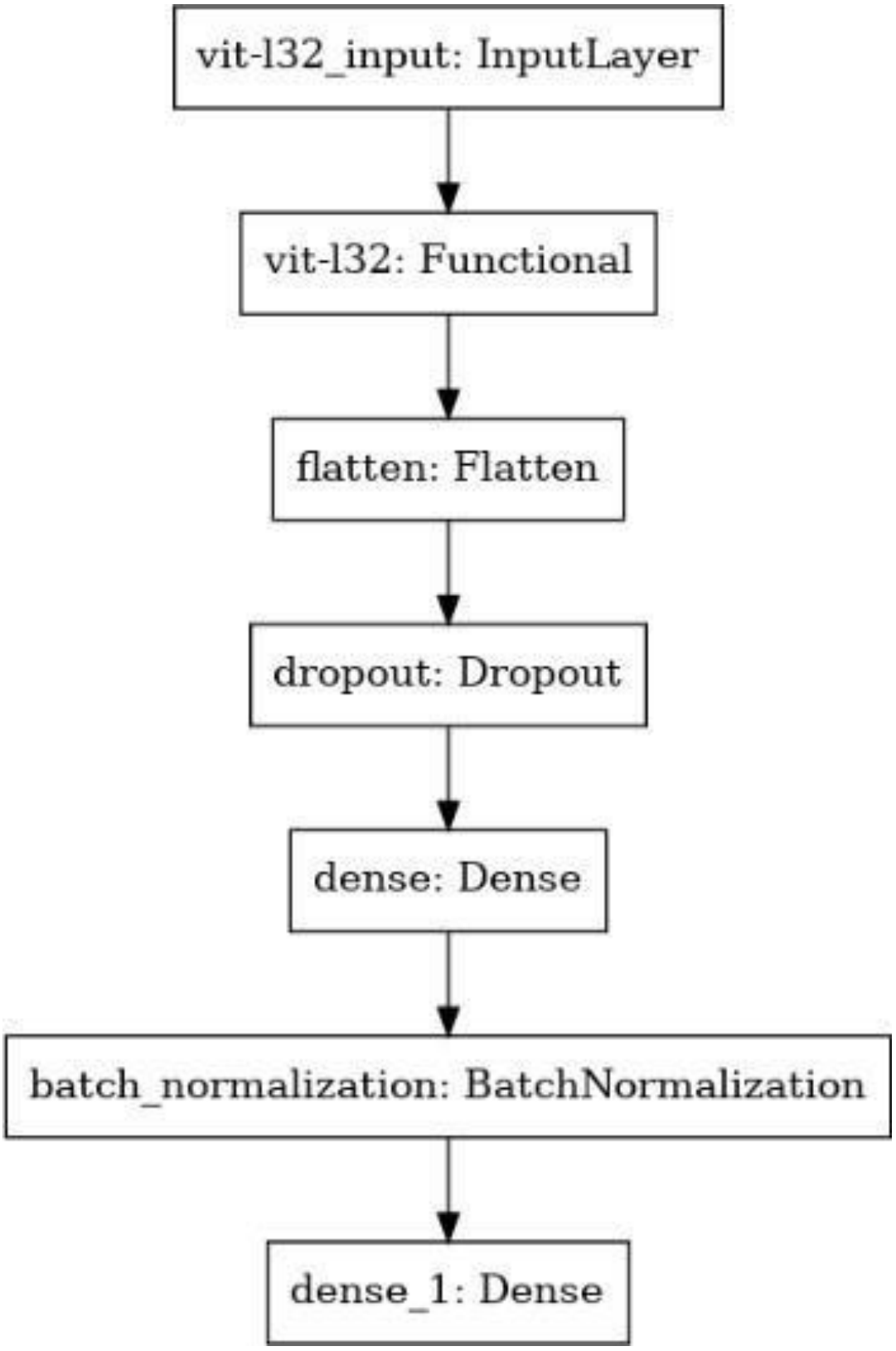

Fig 4.7: Neural Network Layers with ViT_L-32 as a backbone

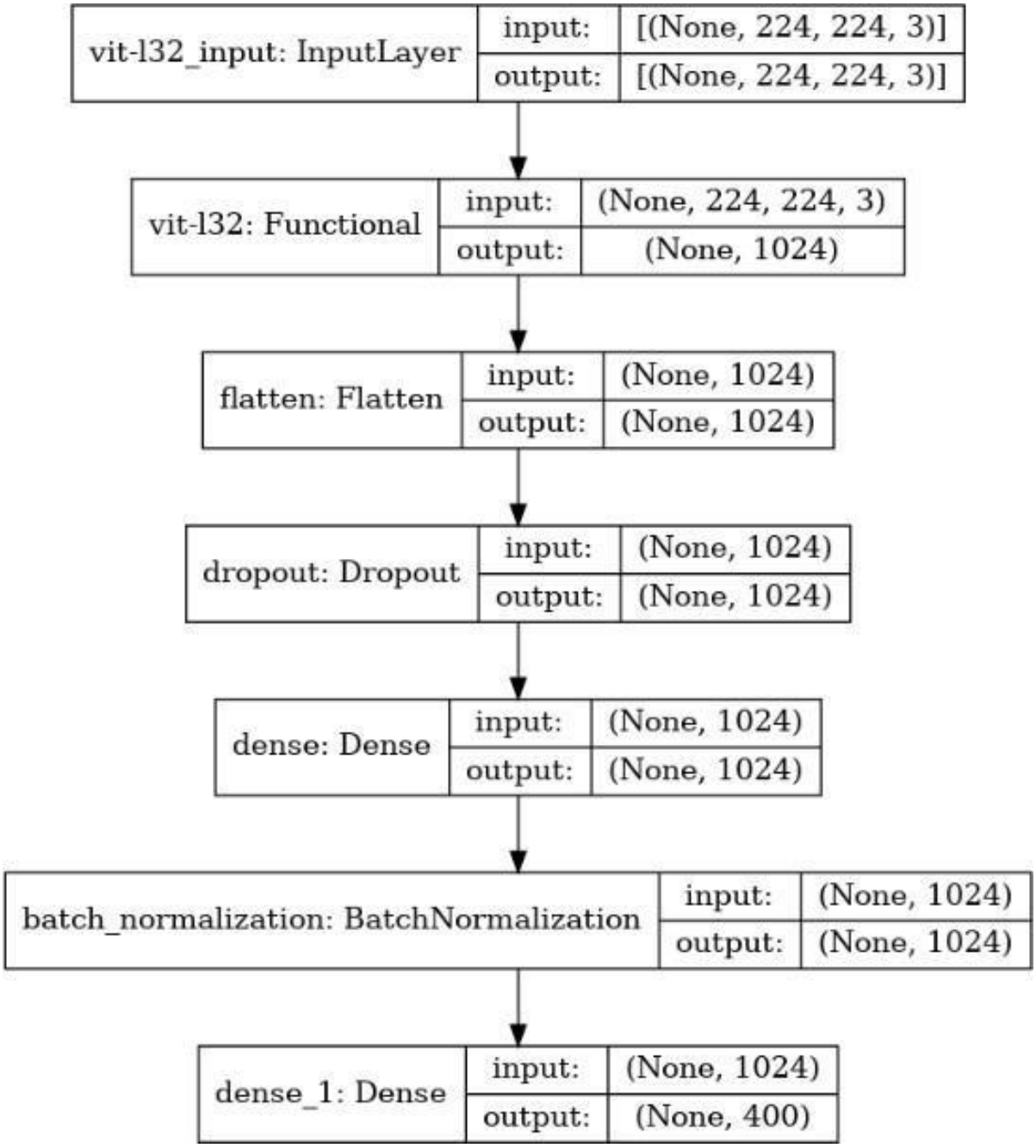

Fig 4.8: ViT_L-32 Neural Network Layers with input and output shapes

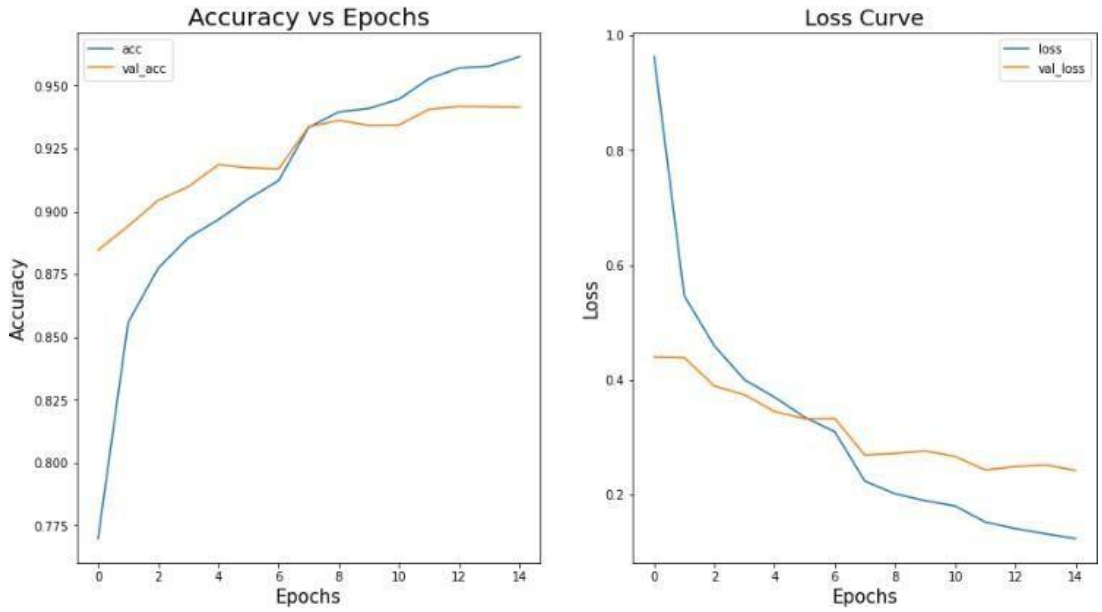

Fig 4.9: Accuracy and Loss Curves with ViT_L-32

Streamlit App for Model Deployment:

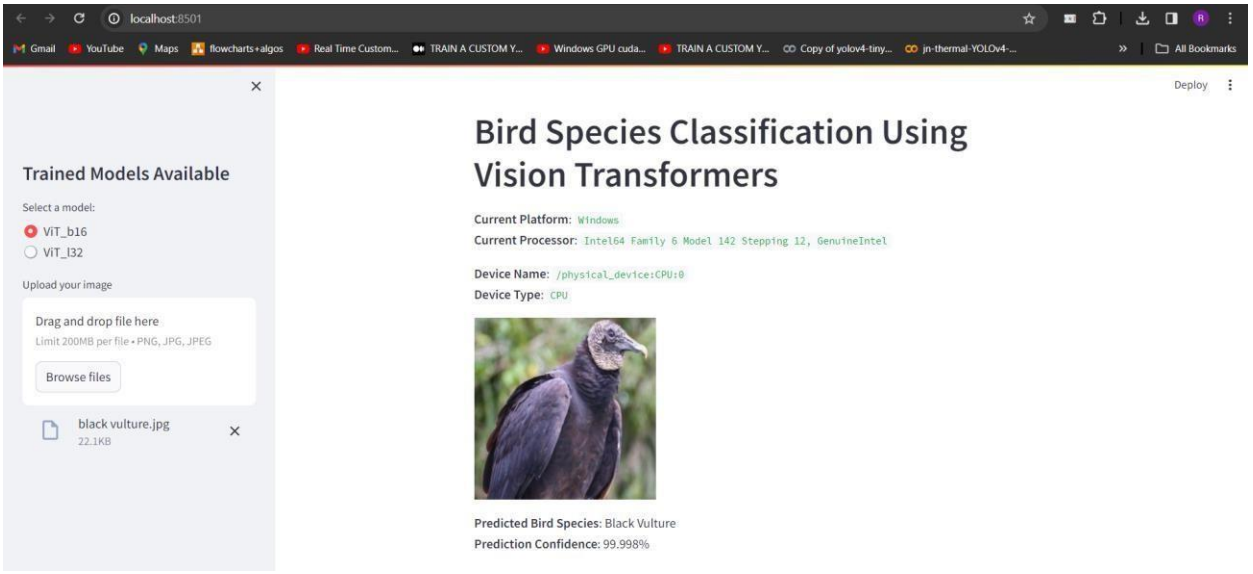

Fig 4.10: Bird Classification with ViT_B-16

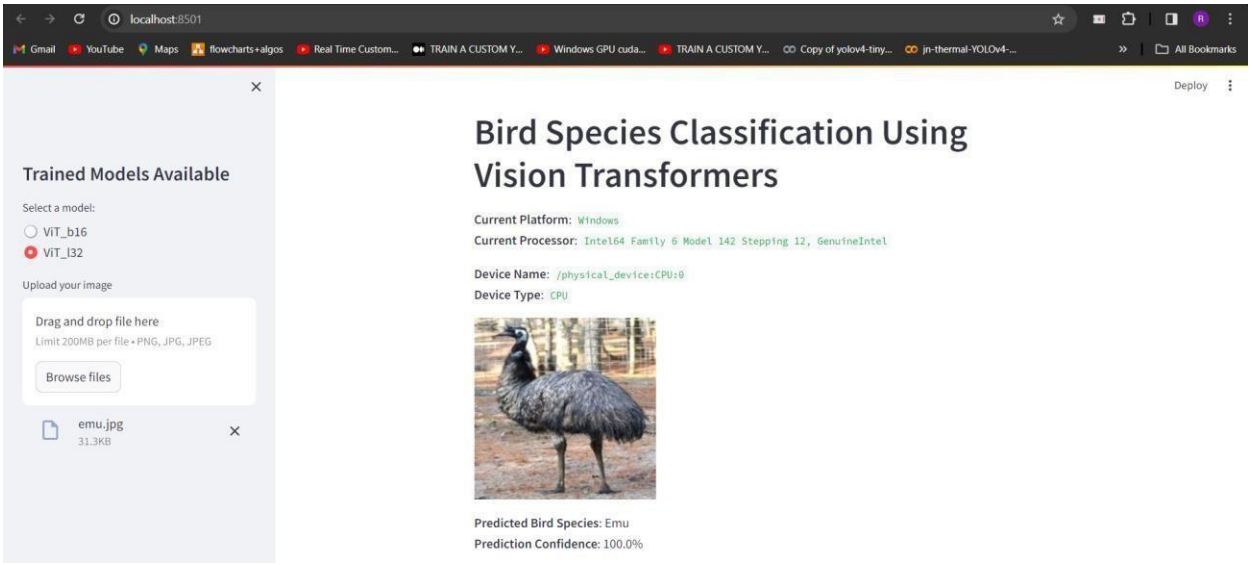

Fig 4.11: Bird Classification with ViT_L-32

4.2 : Software Testing

When evaluating the performance of machine learning models, several metrics are commonly used to assess their effectiveness in classification tasks. These metrics provide insights into different aspects of the model's performance and help in understanding its strengths and weaknesses. In the context of model testing, particularly for image classification tasks using Vision Transformer models, metrics such as F1-score, recall, precision, testing accuracy, and confusion matrix play crucial roles.

- F1-score: The F1-score is a measure of a model's accuracy, balancing both precision and recall. It is calculated as the harmonic mean of precision and recall, providing a single score that represents the model's overall performance. A high F1-score indicates a well-balanced trade-off between precision (the ability of the model to correctly identify positive cases) and recall (the ability of the model to correctly identify all positive cases).
- Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive cases that the model correctly identifies. It quantifies the model's ability to capture all relevant instances of a particular class from the entire dataset. A high recall indicates that the model effectively detects most of the positive cases.
- Precision: Precision measures the proportion of true positive cases among all instances identified as positive by the model. It quantifies the model's ability to avoid misclassifying negative cases as positive. A high precision indicates that the model makes fewer false positive errors.

- Testing Accuracy: Testing accuracy represents the overall correctness of the model's predictions on the test dataset. It is calculated as the ratio of the number of correct predictions to the total number of predictions made by the model. While testing accuracy provides a simple and intuitive measure of model performance, it may not adequately capture the model's performance across different classes, especially in imbalanced datasets.
- Confusion Matrix: A confusion matrix is a tabular representation that summarizes the performance of a classification model on a test dataset. It provides a detailed breakdown of the model's predictions, showing the number of true positives, false positives, true negatives, and false negatives for each class. From the confusion matrix, various performance metrics such as recall, precision, and F1-score can be computed for individual classes, allowing for a more nuanced evaluation of the model's performance across different categories.

4.3 : Experimental Results

Model Name	F1-Score	Recall	Precision	Testing Accuracy
ViT_B-16	0.9560	0.9560	0.9598	95.6033%
ViT_L-32	0.9422	0.9427	0.9464	94.2718%

Table 4.1: Comparison of Model Results

Model Name	Total Parameters	Trainable Parameters	Non-trainable Parameters
Neural Network with ViT_B-16 backbone	87,000,208	1,199,504	85,800,704
Neural Network with ViT_L-32 backbone	306,974,096	1,461,648	305,512,448

Table 4.2: Number of parameters for each model

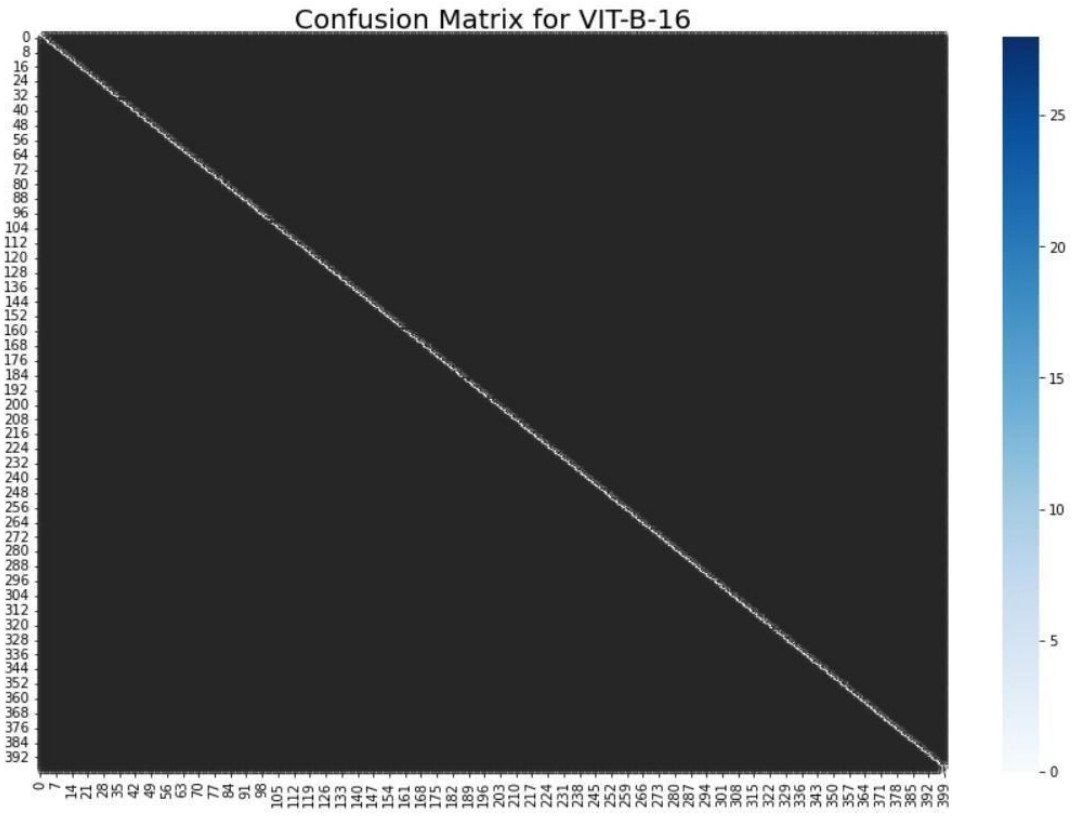

Fig 4.12: Confusion matrix for ViT_B-16

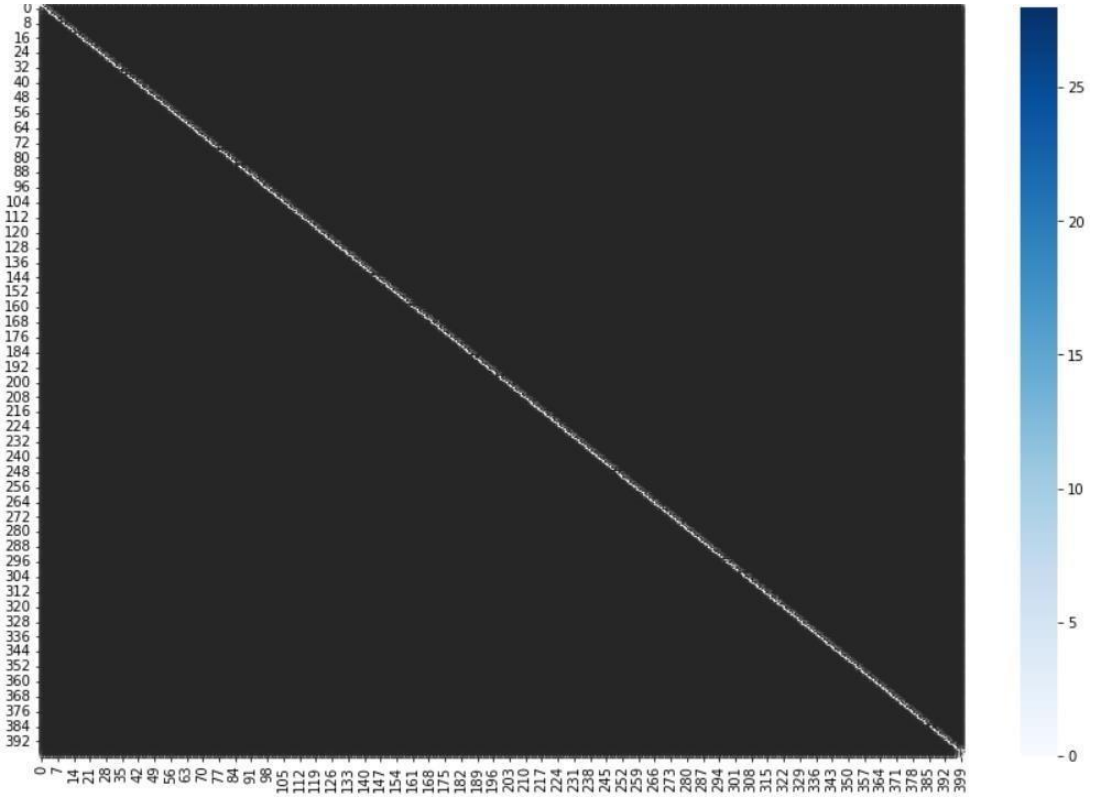

Fig 4.13: Confusion matrix for ViT_L-32

Upon evaluating the models, it is evident that the ViT_B-16 model exhibits slightly superior performance compared to the ViT_L-32 model across multiple evaluation criteria. With an F1-

score of 0.9560, recall of 0.9560, precision of 0.9598, and testing accuracy of 95.6033%, the ViT_B-16 model outperforms its counterpart, the ViT_L-32, which achieved an F1-score of 0.9422, recall of 0.9427, precision of 0.9464, and testing accuracy of 94.2718%. However, the choice between these models necessitates consideration of various factors, including computational resources, training time, and model complexity. Regarding potential enhancements for the ViT_L-32 model through longer training, several strategies could be explored. Extending the training duration allows the model to converge to a more optimal solution by thoroughly exploring the solution space and capturing intricate patterns and representations. Regularization techniques, such as dropout, weight decay, or early stopping, can mitigate overfitting during extended training. Dynamic learning rate scheduling strategies, coupled with data augmentation techniques, can further optimize model performance by enhancing generalization and exploration of the training landscape. Additionally, fine-tuning the ViT_L-32 model's architecture or exploring variations of the Vision Transformer architecture may unlock additional representational power and improve its capacity to capture complex image features. In summary, while the ViT_L-32 model could benefit from longer training durations to enhance its performance, a comprehensive approach that integrates various optimization strategies and architectural modifications is essential for achieving significant improvements effectively.

Chapter 5

Conclusion and Future Work

This chapter presents the conclusion and outlines future directions for the project on bird species image classification. Throughout this endeavor, significant progress has been made in developing an efficient system for detecting and classifying bird species from images. The conclusion summarizes the key findings, achievements, and insights gained from the project, highlighting the effectiveness of the implemented methodologies and the performance of the developed classification system. Additionally, the conclusion reflects on the challenges encountered during the project and discusses potential areas for improvement and refinement.

Looking ahead, the future work section identifies several avenues for further research and development in the field of bird species classification. These include exploring advanced deep learning architectures, such as attention mechanisms and ensemble methods, to enhance classification accuracy and robustness. Furthermore, future work may involve expanding the scope of the classification system to include additional bird species and addressing specific challenges related to rare or poorly represented species. Additionally, efforts could be directed towards optimizing the system's computational efficiency and scalability for real-world deployment in ecological monitoring and conservation applications. Overall, the conclusion and future work sections provide valuable insights and guidance for advancing the state-of-the-art in bird species image classification and fostering continued innovation in this important domain.

5.1 Conclusion and Discussion

In the realm of ornithology and biodiversity monitoring, the project stands as a testament to the fusion of cutting-edge technology and ecological stewardship. The project originated from the recognition of a critical gap in traditional bird species identification methods, prompting the need for a sophisticated solution that aligns with the advancements in deep learning. The problem statement identified the limitations of qualitative descriptions, paving the way for the development of a comprehensive and efficient bird species identification system.

The introduction highlighted the significance of bridging this gap through the utilization of Vision Transformers, advanced deep learning models renowned for their prowess in image feature extraction. By leveraging these models, the project aspires to simplify and enhance bird species identification, contributing to the evolving landscape of ornithological research.

The project's objectives were clearly delineated, encompassing the creation of a robust database, optimization of classifier layers, and the development of a user-friendly web application. The

scope extended to include not only the technical aspects of image processing and model training but also the seamless integration of the system into a user-friendly interface.

The articulated hardware and software requirements set the foundation for the technical infrastructure necessary for the successful implementation of the bird species identification system. From Streamlit for frontend development to TensorFlow and Keras for backend operations, the specified tools and libraries form a cohesive ecosystem, ensuring the efficiency and accuracy of the entire system.

As we delve into future work, the potential expansions for the project are expansive. The integration of real-time identification capabilities into a mobile application opens up new avenues for users, transforming bird identification into a dynamic and interactive experience. Collaborations with conservation organizations provide an opportunity to elevate the project's impact, integrating the identification system into larger conservation efforts and ensuring that the data collected serves a dual purpose – scientific research and species protection.

In conclusion, the project emerges not just as a technological innovation but as a catalyst for citizen science, environmental education, and global collaboration. By addressing the challenges posed in the problem statement, the project moves beyond the confines of traditional ornithology, embracing a future where individuals, armed with mobile devices and a passion for nature, actively contribute to the understanding and preservation of the avian world. As the system evolves and incorporates future enhancements, it is poised to leave a lasting imprint on the intersection of technology and conservation, ushering in a new era of biodiversity monitoring and ecological awareness.

5.2 Scope for Future Work

The scope for future work of the project is as follows:

i. **Enhanced Mobile Application Features:**

- Integration of real-time identification: Enhance the mobile application to provide users with the capability to identify bird species in real-time through the device's camera, leveraging the Vision Transformers for quick and accurate species recognition.
- Offline mode: Develop an offline mode for the mobile application, allowing users to identify bird species even in areas with limited or no internet connectivity. This feature can be beneficial for field researchers and bird enthusiasts exploring remote locations.

ii. **Community-Driven Data Collection:** Encourage users to contribute to bird monitoring efforts by allowing them to submit bird sightings and related data. Implement a community-driven approach to data collection, fostering a sense of participation among users and contributing valuable information for scientific research.

iii. **Conservation Impact Dashboard:** Strengthen partnerships with conservation organizations to integrate a dedicated conservation impact dashboard within the application. This dashboard can showcase the cumulative impact of user contributions on bird species monitoring and protection efforts, promoting transparency and accountability.

iv. **Augmented Reality (AR) Integration:** Explore the integration of augmented reality for an immersive bird identification experience. Users can use their mobile devices to view information about birds in real-world environments, enhancing the educational and interactive aspects of the application.

Bibliograph

[1] Noumida, A., et al. "Stacked Res2Net-CBAM with Grouped Channel Attention for Multi-Label Bird Species Classification." 2023 31st European Signal Processing Conference (EUSIPCO). IEEE, 2023.

[2] A. Marini, J. Facon and A. L. Koerich, "Bird Species Classification Based on Color Features," IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, pp. 4336-4341, doi: 10.1109/SMC.2013.740. keywords: {Birds;Image color analysis;Image segmentation;Feature extraction;Vectors;Visualization;Histograms;pattern recognition;color features;color image segmentation;machine learning;bird species classification}

[3] Asmita Manna, Nilam Upasani, Shubham Jadhav, Ruturaj Mane, Rutuja Chaudhari and Vishal Chatre, “Bird Image Classification using Convolutional Neural Network Transfer Learning Architectures” International Journal of Advanced Computer Science and Applications(IJACSA), 14(3), 2023. http://dx.doi.org/10.14569/IJACSA.2023.0140397

[4] Shriharsha, Tushara, Vijeth, Suraj, Hemavathi, “Bird Species Classification Using Deep Learning Approach”, International Research Journal of Engineering and Technology (IRJET), Volume 07, Issue 04, April 2020.

[5] Samparthi V S Kumar, Hari Kishan Kondaveerti, "A Comparative Study on Deep Learning Techniques for Bird Species Recognition", 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), pp.1-6, 2023.

[6] https://www.kaggle.com/datasets/gpiosenka/100-bird-species

[7] https://paperswithcode.com/method/vision-transformer

[8] https://github.com/faustomorales/vit-keras

[9] https://github.com/google-research/vision_transformer

[10] https://streamlit.io/

Acknowledgement

We express our deepest gratitude to Prof. Kirit Mishra for her exceptional mentorship, invaluable guidance, and unwavering support throughout the implementation of this engineering project. Her profound expertise, patience, and dedication have been instrumental in shaping our perspectives and methodologies.