Small Bird Recognition using Deep Learning: Evolving Strategies for Individual Identification

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Aishwarya Kengal
School of Computer Science
University of Galway

Supervisor(s)
Adrian Clear

In partial fulfillment of the requirements for the degree of
MSc in Computer Science Data Analytics
30-08-2023

DECLARATION I, Aishwarya Kengal, hereby declare that this thesis, titled "Small Bird Recognition using Deep Learning: A Novel Approach to Individual Identification", and the work presented in it are entirely my own except where explicitly stated otherwise in the text, and that this work has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: Aishwarya Kengal

Abstract

Bird identification is fundamental to ecological research, conservation initiatives, and citizen science participation. In this thesis, we propel individual small bird identification to new heights by honing a distinctive approach inspired by a recent study [1]. Leveraging the cutting-edge Detectron framework, methodology achieves unparalleled precision in object detection and localization, significantly elevating the accuracy of bird identification. To fortify the model's robustness, we employ sophisticated data augmentation techniques within the Keras framework, effectively diversifying the training dataset. Moreover, the integration of the VGG19 pre-trained model augments the foundation for fine-tuning, enabling the effective classification of diverse species such as sociable weavers, great tits, and zebra finches. This research not only enhances the performance of the existing model [1] but also introduces a refined paradigm. This meticulous and tailored approach, harmonizing the capabilities of Detectron, data augmentation, and the VGG19 model, represents a significant stride forward in individual bird identification. Beyond the immediate impact on individual identification accuracy and efficiency, this novel combination holds broader implications for avian conservation and ecological understanding.

Keywords: Small birds, Individual Identification, Deep Learning, Convolutional Neural Networks, Transfer Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past, traditional methods such as banding and tagging on the birds have been widely used but faced limitations that hindered widespread adoption. However, a previous project [1] has already proposed a novel methodology for precise identification of birds based on their unique visual characteristics. They addressed key challenges such as data pre-processing, feature extraction, model training, and system evaluation. The researchers have worked with a major bird's dataset that includes sociable weavers, great tits, and zebra finches. They have successfully developed a model that can accurately identify birds based on their visual characteristics. Special attention has been given to ensuring that the model has the ability to generalize across new images, diverse lighting conditions, different backgrounds, and various bird orientations. The researchers have already implemented strategies to train the model using a diverse range of bird images, accounting for different environmental factors and bird poses.

By building upon the groundwork established by the prior project [1], this thesis strives to enrich the realm of bird identification through exploration of data preprocessing methodologies. This endeavor seeks to impart meaningful contributions to the domains of ecological research, conservation endeavors, and citizen

science initiatives. To achieve this objective sequential steps taken to address the task.

1. Leveraging Detectron2 for Data Pre-processing: Focused on advancing bird identification methods by leveraging the capabilities of Detectron2 [2], a state-of-the-art deep learning framework known for its image segmentation and object detection. With the integration of Detectron2, I successfully pre-processed the bird dataset, generating high-quality cropped bird images and standardizing image properties while eliminating noise. This data preparation laid a robust foundation for achieving exceptional model performance. Detectron2's robust object detection algorithms enabled accurate identification of individual birds, even in challenging scenarios involving occlusion and partial visibility. Its exceptional performance speed, GPU acceleration, and efficient implementation of convolutional neural networks (CNNs) further ensured rapid inference, reducing processing time. During the experimentation phase, I also explored the application of YOLO [3, 4] segmentation for data pre-processing. However, the results were not as promising as with Detectron2. YOLO struggled to handle the complexities present in the bird dataset, leading to suboptimal performance and less accurate instance segmentation. Consequently, I opted to focus on utilizing Detectron2 due to its superior performance, robustness, and adaptability to the specific challenges of the bird identification task.

2. VGG19 Pre-trained model for Enhanced Accuracy: Feature extraction plays a crucial role in distinguishing unique visual characteristics among the individual bird. I employed VGG19 [1], a pre-trained convolutional neural network, for model training with data augmentation. I fine-tuned the VGG19 model by introducing dropout regularization and adding a fully connected layer with a SoftMax output of classes (specifying different individual birds

of same species). The model was compiled using the Adam optimizer with a learning rate of $1^{(e-5)}$ and the categorical cross-entropy loss function. The training data was augmented using random rotations and zooming to increase the diversity of the dataset and improve model generalization. The training process was monitored using the Model Checkpoint callback to save the best model based on the lowest loss value, and Early Stopping callback to stop training if there was no improvement in the model for three consecutive epochs. The model was trained for ten epochs using a batch size of eight, and the performance was evaluated on the validation data. The training process aimed to maximize the potential of the pre-trained VGG19 model on the bird dataset, considering diverse bird species, lighting conditions, and backgrounds to achieve a more refined and accurate model.

3. Evaluation and Validation: Using the preprocessed testing images, I employed the trained VGG19 model to predict the identities of the individual birds in the pictures. The model extracted features from each image and made predictions based on those features, identifying the corresponding individual bird. The evaluation is done by calculating the accuracy and comparing the model's predicted identities with the actual identities from the testing dataset.

1.1 Motivation

My deep-rooted fascination with nature and animals, coupled with a desire to contribute to ecological research and conservation, has driven my quest for a meaningful thesis topic. Through discussions with my supervisor and interactions with a PhD student, I discovered the realm of citizen science projects and the potential for individual bird identification and tracking. Encountering [1] which

showcased a ground-breaking methodology for precise bird identification, sparked my enthusiasm further. Inspired by this research, I am motivated to expand upon the existing techniques, address challenges, and contribute to the advancement of bird identification. Through my thesis, I aspire to make a positive impact on ecological studies, engage citizen scientists, and foster a greater understanding and conservation of bird species and their habitats.

1.2 Research Questions

1. How does the implementation of YOLOv8-based object detection and the utilization of the Detectron framework for data preprocessing enhance the accuracy and effectiveness of bird recognition and individual tracking, in comparison to conventional approaches, and what insights can be gained from this exploration?

2. The Convolutional Neural Network (CNN), specifically the VGG19 model used in the previous study [1], has shown promising results in identifying and tracking individual birds. Building on these results, this project will fine-tune the hyperparameters of the VGG19 model to further enhance its performance and adapt it to this specific task.

Chapter 2

Background

2.1 What is deep Learning and why to use deep learning?

Deep learning is a subfield of machine learning that has gained significant traction in ecology, especially for tasks like animal or plant species identification and enumeration from images [1]. The fundamental idea behind deep learning is to mimic the way humans acquire knowledge by automatically learning relevant features from the data. This ability to learn intricate patterns and representations from large datasets makes deep learning a powerful tool for various domains [5]. One of the key advantages of deep learning is transfer learning, which allows us to leverage pre-trained models for new tasks. For instance, in [1], the researchers utilized the pre-trained VGG19 model to extract essential features such as color, texture, and patterns, enabling effective differentiation between multiple objects, which in turn facilitated the identification of individual birds with high accuracy. The versatility and capability of deep learning to automatically learn complex representations make it an invaluable tool for advancing ecological research and

conservation efforts.

2.2 Convolutional Neural Network

Individual Identification plays a crucial role in ecological research. Deep learning methodologies, including convolutional neural networks (CNNs), have garnered interest among ecologists [1]. Deep learning uses CNN, a network architecture that derives its knowledge directly from images. It is made up of several layers that process and transforms the input to produce an output. CNN can be trained for Image classification, image segmentation, object detection. Each neuron in the input layer and each neuron in the hidden layer are connected to one another in the basic neural network architecture. However, in the CNN, only a little portion of the input layer's neurons are linked to a tiny portion of the hidden layer's neurons; this tiny portion is known as the local receptive field. The local receptive field is translated across an image to create a feature map from the input layer neurons to the hidden layer neurons. For all hidden neurons in the specified hidden layer, the shared weights and bias values are the same. This makes the network robust to translation of the object in an image because all hidden neurons are detecting the same feature, such as an edge or a blob, in various regions of the image. The activation and pooling functions come next. Utilizing activation functions (ReLU), the activation stage applies the transformation to each neuron's output. It takes the neuron's output and maps it to the largest positive value, or zero if the neuron's output is negative [6].

2.3 Data collection approaches and its importance

Data plays an essential role in the characterization, calibration, verification, validation, and evaluation of models used to predict the long-term durability and performance of materials in harsh environments [7]. Many models would be useless without sufficient data to validate and evaluate them. A database containing many master pedigrees of material systems and a standard dataset that represents a variety of test settings and applications is necessary for an accurate structural durability model.

Camera trapping is a method used to capture wild animals which provide data on species location, population sizes and how species are interacting when researchers are not present. This is being used in the ecological research for decades [8]. In March 2006, they [8] set up a grid of 12 camera trapping stations at the Jasper Ridge Biological Preserve (JRBP) in California, USA. The stations were being operated continuously for 2 years. They collected the film roll from camera trap and stored all the data in their database, they considered using only the bobcat images (1072 images) for individual identification. To identify individual bobcats, they first grouped the pictures into clusters based on the camera-trap station, date, and time of their occurrence. By applying 3min limit they could generate 487 picture time clusters. They labelled the individual bobcats through online web interface which they created. The study aimed to assess the agreement between two classifiers in naming picture clusters and evaluate if the number of pictures in a cluster influenced the probability of them being named. The level of agreement between classifiers was measured using the adjusted Rand index (ARI), which compares the similarity of two partitions of picture clusters. The ARI takes into account chance agreement and has a range from zero to

one, with higher values indicating greater similarity. The ARI was calculated at three different stages: after the first round of naming, after one classifier used the reconciliation tool, and when both classifiers finished using the tool. Monte Carlo simulations were also conducted to compare the consensus reached by the classifiers with random processes. The size distribution of named picture clusters was compared to the initial distribution to examine size-based biases in cluster selection. By analyzing these aspects, the study aimed to provide insights into the agreement between classifiers and potential biases in the naming process based on cluster size.

Authors [1] have used a RFID based camera trap, in order to make use this there should be a communication established between the RFID logger and the Raspberry Pi camera. The RFID data logger detected birds and sent their PIT-tag codes to the Raspberry Pi, which took pictures at regular intervals. To avoid overfitting of the CNN model they avoided collecting pictures with near identical frames. Each picture file was labelled with the bird's identity from the RFID logger at the time of taking a picture. This automated process eliminated the need for manual identification and annotation. Pictures with multiple birds were automatically excluded from the dataset.

Citizen science and crowdsourcing which involves in data collection uses multiple sources, collecting data themselves, adapting old data, sharing, or exchanging the data, purchasing the data. When there is good amount to data, we can get better insights and patterns to utilize and deal with it.

2.4 Data Pre-processing Techniques

Object detection, a crucial task in computer vision, has been revolutionized by the emergence of deep learning, and the integration of frameworks like Detectron has

further advanced the field. Object detection methodologies can be broadly categorized into two types: two-stage detectors and single-stage detectors. Two-stage detectors, exemplified by R-CNN and its variants, utilize complex architectures and selective region proposals to first identify regions of interest and subsequently classify those regions. These methods are renowned for their high detection accuracy but tend to be slower due to their multi-step approach.

In contrast, single-stage detectors, exemplified by You Only Look Once (YOLO), adopt a simpler architecture and aim to detect objects across all spatial regions in a single pass. While traditional single-stage detectors may have sacrificed some accuracy compared to two-stage detectors, they excel in terms of significantly faster inference times, making them highly suitable for real-time object detection tasks.

The incorporation of Detectron [2], a state-of-the-art deep learning framework, has further elevated the performance of single-stage detectors. Detectron and its variants have made significant strides in enhancing detection accuracy, often surpassing the performance of two-stage detectors. For instance, when compared to Fast-RCNN, a two-stage detector achieving 70% detection accuracy, YOLO, despite having a slightly lower accuracy of 63.4%, exhibits a remarkable approximately 300-fold improvement in inference speed [3]. This integration of Detectron has proven to be a game-changer in the domain of object detection, empowering researchers and practitioners with both accuracy and efficiency to tackle diverse and challenging computer vision tasks.

Detectron's [2] prowess in object detection and instance segmentation finds an impactful application in the domain of avian research, specifically in identifying and segmenting birds within the images. Leveraging Detectron for bird identification involves a series of steps that synergistically combine its advanced algorithms to yield accurate and finely segmented results. The process begins with the inte-

gration of Detectron's pre-trained models, such as the Mask R-CNN [9], tailored to object detection and instance segmentation tasks. When applied to bird images, Detectron initially processes the entire scene to detect potential regions of interest where birds might be present. These regions are proposed based on the detected features and are refined using subsequent layers of the network. Once regions of interest are identified, Detectron employs its instance segmentation capabilities to meticulously segment each detected bird instance. This involves not only outlining the boundaries of the birds but also pixel-wise classifying each pixel as belonging to a particular bird. This high-resolution segmentation map provides a detailed understanding of the spatial distribution of bird instances within the image. One of the remarkable aspects of Detectron's segmentation is its ability to achieve precise boundaries, even in challenging scenarios such as occlusion or overlapping instances. This is particularly relevant in bird identification, where accurate segmentation is crucial for distinguishing individuals of the same species within a crowded scene. Importantly, Detectron's segmentation output includes pixel-level masks that delineate the exact outline of each bird. These masks serve as powerful tools for generating cropped images of individual birds. By applying the pixel-wise masks to the original image, the corresponding bird instance is isolated from the background, resulting in a high-quality cropped image. These cropped images are invaluable for subsequent analysis and model training, ensuring that only relevant and noise-free data are used. In summary, Detectron's utilization for bird identification involves leveraging its robust object detection and instance segmentation capabilities. This process results in precise identification of bird instances within images, accompanied by detailed pixel-level segmentation masks. By applying these masks, the images can be efficiently cropped, providing a valuable dataset for further analysis, model training, and ecological research. Detectron's versatility, accuracy, and efficiency make it an

indispensable tool in advancing avian research and conservation efforts.

Chapter 3

Related Work

The following section contains topics that are worth considering for individual identification in small birds using deep learning.

Machine learning's evolution from pattern recognition to the dominant approach in data analytics is central[10] to this project. It forms the core of the bird identification system, allowing the identification of birds based on their unique visual characteristics. Using a pre-trained VGG19 model, we tap into the benefits of transfer learning, fine-tuning the model to specific task. The accurate outcomes produced will aid in ecological research and conservation efforts, highlighting the pivotal role of machine learning in the project's success.

3.1 Summary of base papers

Ferreira et al.,2020 [1] considered using the VGG19 CNN architecture. Freezing the lower layers of the network is a common practice in transfer learning to prevent overfitting, especially when dealing with limited training data. However, they opted not to freeze these layers because doing so could impede the acquisition of critical features essential for precise classification due to size of their training

dataset. Instead, they decided to replace the classifier part of the VGG19 CNN network, known as the fully connected part, with new layers that had randomly initialized weights. These new layers were designed specifically for their task of interest, which was individual bird recognition, taking into account the number of different individuals they wanted to classify. This approach allowed to make use of the learned features from the pre-trained network while customizing the classifier for their specific task of identifying individual birds. One dropout layer was added before the first dense layer to avoid overfitting by randomly ignoring units of the CNN during the training process. As the model did not improve on the accuracy when the dropout was at initial 0.5 and the random guess of epochs was 10, the authors used ADAM optimizer with the learning rate $1^{(e-5)}$ and the SoftMax activation function to the classifier. If there was no decrease in the loss they retrained the model with different optimizer and changing the learning rate until they achieved low loss. When model analysed the results for different birds they identified that it is crucial to investigate the distortion level at which the performance of the network begins to decrease, and to check whether the structure of the network significantly affects its ability to be invariant to quality distortions so they used the transformation techniques such as Gaussian blur, motion blur, Gaussian noise, resizing transformations and a random combination of two of these four transformations to lower the quality of the images in the dataset. To test their model they captured images with different cameras and different perspectives to test whether their CNN achieved in identifying individuals without overfitting to their training dataset. The model was able to achieve 92.4 accuracy for sociable weavers after training for 21 epochs, 90% accuracy for great tits after training for 32 epochs and 87% accuracy for zebra finches after training for 11 epochs. The major challenge they discussed is the applicability of CNN is difficult while considering to collect the data of different species in long term since the

appearance and the size of the species might have changes with time eventually and this will be a research problem for ecologists.

In their [11] experiment the dataset included 65000 face images of giant panda, they have used VGGNet which consists of 5 convolution modules, 3 fully-connection layers, and a soft-max layer. Each set of convolution modules consists of a convolutional layer and a pooling layer. The convolutional layer extracts image features, while the pooling layer reduces the image's complexity by retaining key features. After convolution and pooling, the extracted features are classified using the fully-connected layers. Lastly, the SoftMax layer performs probability mapping, selecting the category with the highest probability [12]. They [11] made changes to the network according to their dataset by replacing the last pooling layer with a Spatial Pyramid Pooling Layer (SPP). To train the model easily and to avoid the Exploding Gradient Problem(EGP) they replaced the Dropout layer with the batch normalization layer by reducing the number of network layers to 11. Their dataset of giant panda which then was divided to training, validation and testing. They considered few calibration images to set the hyperparameters. To test the model in the field they used image transformation techniques like to clean and blurring/dirtying(saturation levels) the images and used face rotation. They used multi-way ANOVA and t-test to compare the results. Their model achieved 95.0% accuracy in identifying the general test set.

The researchers [13] proposed a system to assist in the identification of elephants using image data. The approach involved several steps: Object Localization: They used a pre-trained YOLO (You Only Look Once) network to automatically locate the heads of elephants in images. The network was trained on a separate dataset of elephant images from Flickr. The results showed a precision of 92.73%, recall of 92.16%, and mean average precision of 90.78% for head detection.

Bounding Box Correction: The user had the option to correct the automatically predicted bounding boxes by drawing new ones or selecting from multiple proposed boxes. This step ensured accurate localization of the elephant heads.

Feature Extraction: The selected bounding boxes were cropped and fed into a modified ResNet50 network to extract features. The network was trained on the ImageNet dataset and modified to extract features from earlier activation layers, which were followed by a new pooling layer for increased translation invariance.

Dimensionality Reduction: To handle the high dimensionality of the extracted features, principal components analysis (PCA) was applied to reduce the number of features to twice the number of training images.

Classification: A support vector machine (SVM) was used to classify the extracted features. The individual elephants (classes) were ranked based on their confidence values obtained from the SVM.

Analysis and Results: The researchers evaluated the performance of their system using different configurations. They found that using the activation layer of the 14th residual block in the ResNet50 network provided the best results for feature extraction. By adding a pooling layer, even better results were achieved. The top-1 accuracy ranged from 52.4% to 56%, and the average per-class accuracy was 49% for single-image classification. When using two images for classification, the top-1 accuracy increased to 70.8% and the average per-class accuracy to 59%.

Table 3.1: Literature Study Overview table

Paper + Dataset	**Model Used**	**Comments**	**Results**
Ferreira et al., 2020 [1] used Birds dataset	VGG19 CNN	Freezing lower layers vs. replacing classifier. Dropout layer to avoid overfitting. Transformation techniques to lower image quality. Challenges of long-term data collection for different species.	92.4% accuracy for sociable weavers after 21 epochs. 90% accuracy for great tits after 32 epochs. 87% accuracy for zebra finches after 11 epochs.
Jin Hou et al., 2020 [11] used Giant Panda dataset	VGGNet	Introduction to VGGNet architecture. Replacing pooling layer with Spatial Pyramid Pooling Layer. Replacing Dropout layer with batch normalization layer.	95.0% accuracy in identifying general test set.
Matthias Körschens et al., 2018 [13] used Elephant dataset	YOLO + ResNet50	Object localization with YOLO network. Bounding box correction for accurate localization. Feature extraction with modified ResNet50 network. Dimensionality reduction with PCA.	Precision: 92.73%, Recall: 92.16%, mAP: 90.78% for head detection. Top-1 accuracy: 52.4% to 56% for single-image classification. Average per-class accuracy: 49% for single-image classification. Top-1 accuracy: 70.8%, Average per-class accuracy: 59% with two images for classification.

Chapter 4

Methodology

4.1 Dataset

The dataset comprises three distinct categories of bird images: sociable weavers, great tits, and zebra finches. The labeling process was conducted using different techniques, with wild birds, such as sociable weavers and great tits, being labeled separately from captive birds, specifically zebra finches. The imagery of the latter was captured utilizing raspberry pi cameras [1]. In the wild setting, the data collection involved the placement of an RFID antenna in proximity to bird perches. Seeds were positioned to attract the birds, and an RFID data logger, connected to the raspberry pi camera, was programmed to trigger image capture upon detecting the birds and receiving the corresponding PIT-tag code for each individual. As for the zebra finches, they were observed in a non-socially isolated cage, with raspberry pi cameras capturing images at a rate of one picture every 2 seconds for a group of 10 zebra finches [1].It has been organized into training, testing, and validation sets, each residing in separate folders. Within these folders, images of individual birds are further categorized by their unique identification.

Great tits: The great tits subset of the dataset encompasses extensive training data for each of the 10 individual birds, encompassing 600 images per individual. Alongside, a validation set has been assembled, incorporating 100 images for each individual, serving as a reference for evaluating model performance. Furthermore, a test subset of 50 images per individual has been reserved to comprehensively assess the model's ability to generalize across diverse scenarios.

Zebra Finches: For the zebra finches' subset, an extensive training dataset has been carefully assembled, encompassing a considerable 1200 images for each individual bird. Furthermore, a distinct validation set, consisting of 100 images per individual from the same species, has been prepared to assess the model's efficacy during the training process. Additionally, the selected test subset containing 50 images per individual has been reserved to evaluate the model's ability to generalize across the images. This dataset includes a total of 10 distinct individuals, ensuring a robust representation of the zebra finches' population.

Sociable weavers: The sociable weavers' segment of the dataset is characterized by a training component, encompassing a substantial collection of 800 images for each individual bird. This abundant training data facilitates robust learning and feature extraction, enabling the model to discern distinctive visual attributes specific to each bird. Additionally, a dedicated validation subset has been thoughtfully established, encompassing 100 images for each of the 30 individual sociable weavers. This validation set plays a pivotal role in assessing the model's performance and generalization capabilities, serving as a benchmark to gauge its accuracy and effectiveness. Notably, the testing approach for sociable weavers diverges from the conventional image dataset. As a result of the image acquisition process, which involved extracting images from videos, the testing dataset exclusively comprises cropped images [1].

This structured arrangement enables efficient data handling and ensures that

each bird's visual characteristics are accurately captured and distinguished for subsequent analysis and model training.

(a) Great Tits

(b) Sociable weavers

(c) Zebra Finches

Figure 4.1: Original images collected in the wild and capitive are taken from [1]

4.2 Data Cleaning

Certainly, data cleaning is an essential step in preparing the dataset for training. Since the dataset encompasses images of different individual birds from the 3 species, habitats, and contexts, there's a possibility of containing noise, inconsistencies, and irrelevant images.It's important to ensure that the data fed into the model is accurate, relevant, and representative of the task at hand.

Before proceeding with data preprocessing, a critical data cleaning phase was executed to enhance the quality and suitability of the dataset. Firstly, instances where multiple birds were present within a single image were identified and excluded from the dataset[1]. This step aimed to eliminate potential ambiguities that might arise from images containing more than one individual. By focusing exclusively on images with a single bird, the dataset was refined to facilitate precise individual recognition. Additionally, images depicting birds that were partially obscured or overlapped with other objects were carefully examined. Recognizing the importance of accurate feature extraction and segmentation, images that posed challenges in cropping and isolating the exact bird of interest were excluded. This strategic curation aimed to optimize the training process by providing clear and well-defined images, essential for the subsequent data preprocessing techniques. By undertaking these data cleaning measures, the dataset was thoughtfully curated to ensure that it consisted of high-quality images suitable for effective preprocessing and subsequent model training.

The final dataset used for training, testing, and validation encompassed distinct quantities of images for each species, following a selection process that excluded certain images based on above mentioned criteria:

1. Great Tits: Out of the original 7,733 images, a total of 7,455 images were retained.

2. Sociable Weavers: Out of the original 27,101 images, a total of 24,099 images were retained.

3. Zebra Finches: Out of the original 17,859 images, a total of 14,439 images were retained.

These carefully curated and sizable datasets served as the backbone for the subsequent stages of the research, including preprocessing, model training, and performance evaluation.

Figure 4.2: Images with more than one bird

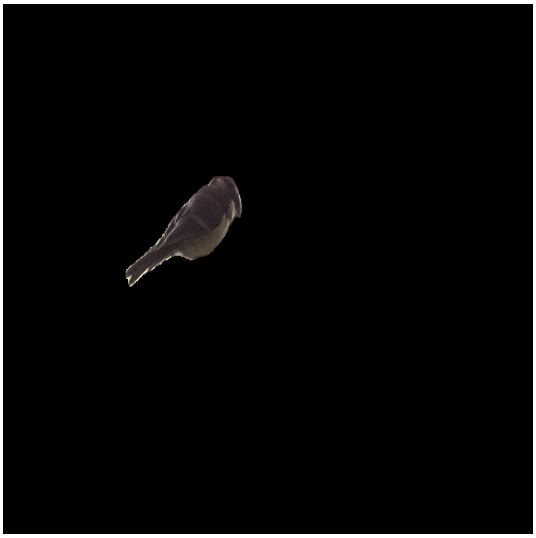
4.3 Data pre-processing using yolo segmentation and the error rate.

The selection of the YOLOv8 architecture for data preprocessing was driven by a strategic consideration of maximizing the probability of success in the task of individual bird recognition. YOLOv8 was chosen based on its reputation as a

cutting-edge and highly effective object detection and segmentation model. This choice was rooted in its potential to offer significant advantages and improvements over other approaches. One key factor that influenced the decision was YOLOv8's position as a presumed state-of-the-art architecture [14]. It was perceived as being at the forefront of object detection and segmentation capabilities, its impressive performance metrics, such as higher mean average precision (mAP) scores. The mAP metric reflects the accuracy and robustness of a model in detecting and localizing objects within an image. YOLOv8's superior mAP scores indicated its proficiency in accurately identifying objects, which aligned well with the objective of precisely isolating individual bird instances within images. Moreover, YOLOv8 was also recognized for its relatively lower inference speed on benchmark datasets like COCO (Common Objects in Context) [14], which implied efficient processing and the potential for faster data preprocessing. This efficiency was of paramount importance, especially when dealing with large datasets. By leveraging YOLOv8 for data preprocessing, we aimed to capitalize on its advanced capabilities, anticipating that it would excel in accurately segmenting and cropping bird instances from original images. The assumption was that YOLOv8's prowess in object detection, coupled with its efficient processing speed, would streamline the creation of high-quality training data. However, through experimentation, it was discovered that the model's accuracy and consistency in detecting and segmenting birds exhibited variability across different images. This variability prompted a reevaluation of the model's suitability concerning the specific task and characteristics of the dataset.

In the specific context of utilizing YOLOv8 segmentation for data preprocessing, a detailed procedure was followed to extract bird images. This procedure involved loading and resizing each image (640,480) [15], generating predictions using the YOLOv8 model, isolating bird masks, determining bounding box coor-

dinates, and ultimately extracting individual bird images. However, despite the systematic approach, the reliance on the accuracy of YOLOv8 model predictions and the quality of generated segmentation masks introduced certain limitations. Notably, the error rate in terms of missed detections and inaccurate segmentation became apparent when processing images for each species. For the Great Tits species, out of a total of 7,733 original images, YOLOv8 segmentation resulted in 6,271 cropped images. This indicates an error rate of approximately (7,733 - 6,271) / 7,733 = 18.8% in the segmentation process. Similarly, for Zebra Finches, from the original 17,859 images, 11,805 cropped images were obtained, indicating an error rate of approximately (17,859 - 11,805) / 17,859 = 33.8%. In the case of Sociable Weavers, out of the 27,101 original images, YOLOv8 segmentation yielded 20,406 cropped images, resulting in an error rate of approximately (27,101 - 20,406) / 27,101 = 24.6%. These error rates reflect instances where the YOLOv8 segmentation approach failed to accurately identify and extract birds from the images, thus contributing to the limitations observed in this data preprocessing technique. This analysis underscores the importance of addressing such errors to enhance the reliability of subsequent model training and overall research outcomes. In summary, the choice of YOLOv8 for data preprocessing was motivated by its reputation as a state-of-the-art model with high mAP scores and efficient inference speed [14]. It was anticipated that YOLOv8's capabilities would align well with the goal of accurately segmenting individual bird instances. Despite this initial assumption, the subsequent evaluation revealed insights into the model's limitations for the specific task, prompting us to explore alternative strategies to achieve the desired level of data quality and reliability.



(a) Great Tits

(b) Sociable weavers

(c) Zebra Finches

Figure 4.3: Images Cropped using Yolov8 segmentation

4.4 Data pre-processing using Detectron2

The choice to transition to the Detectron framework was motivated by its reputation as a robust and versatile toolkit for computer vision tasks, particularly for instance segmentation [16]. Detectron's architecture and methodology are well-

regarded for their ability to achieve fine-grained and precise object segmentation, which aligned seamlessly with the objective of isolating individual bird instances from complex backgrounds.

1. **Configuration Setup:** The data preprocessing commences with a meticulous configuration setup. Leveraging the `"get_cfg()"` function [2], the Detectron2 configuration is initialized. This configuration setup entails the utilization of suitable GPU resources to leverage the capabilities of the Detectron2 model effectively. The configuration is seamlessly merged with the specific configuration file tailored for instance segmentation in this instance, the `"mask_rcnn_R_50_FPN_3x.yaml"`. This pivotal step ensures that the preprocessing pipeline is aligned with the requirements of the instance segmentation model. Crucially, the detection threshold for the model is fine tuned to 0.5, striking an optimal balance between detection precision and recall. Furthermore, the model's weights, integral to its performance, are loaded from the Detectron2 model zoo. This adept configuration setup establishes the groundwork for subsequent image processing and instance identification.

2. **Image Loading and Processing:** The subsequent phase immerses into the loading and processing of images, constituting a foundational stride within the preprocessing pipeline. Functioning within an iterative loop, the pipeline consecutively handles each image extracted from a designated roster of image paths. A pivotal facet of this phase revolves around the images' original dimensions, as they are maintained without alteration. This conscious decision stems from the recognition of each image's unique attributes and the desire to preserve them authentically. The unaltered images then proceed through the Detectron2 predictor, embarking upon the intricate journey of instance identification and segmentation.

3. **Instance Identification and Segmentation:** The core of the preprocessing pipeline lies in accurate instance identification and segmentation. The output generated by the predictor is harnessed to pinpoint instances of bird subjects within the images. A decisive filtering step ensues, focusing solely on instances belonging to the `"bird"` class. This filtering operation optimally selects bird instances, preparing them for subsequent cropping and masking procedures. Each instance's critical attributes notably, its bounding box coordinates and segmentation mask are captured, forming the foundation for precise image cropping.

4. **Cropping and Masking:** The identified instances undergo a transformative cropping and masking procedure. Employing the bounding box coordinates as guides, the pipeline performs precise cropping of the images to extract isolated bird instances. A crucial facet of this step is the creation of a binary mask derived from the segmentation mask. This binary mask serves as a blueprint for effectively isolating the bird region from the background, laying the groundwork for subsequent image refinement.

5. **Grayscale and Thresholding:** Moving towards additional enhancement, the isolated bird region undergoes a conversion to grayscale. Converting to grayscale can help reduce the impact of noise and variations in color, which can be especially beneficial for enhancing image quality and feature extraction [17]. Following this, a deliberate thresholding operation is conducted, leading to the formation of a binary mask. This binary mask plays a crucial role, clearly demarcating the boundary separating the bird from its environment a fundamental step before the ensuing application of the mask.

6. **Mask Application:** The binary mask plays a central role in the mask

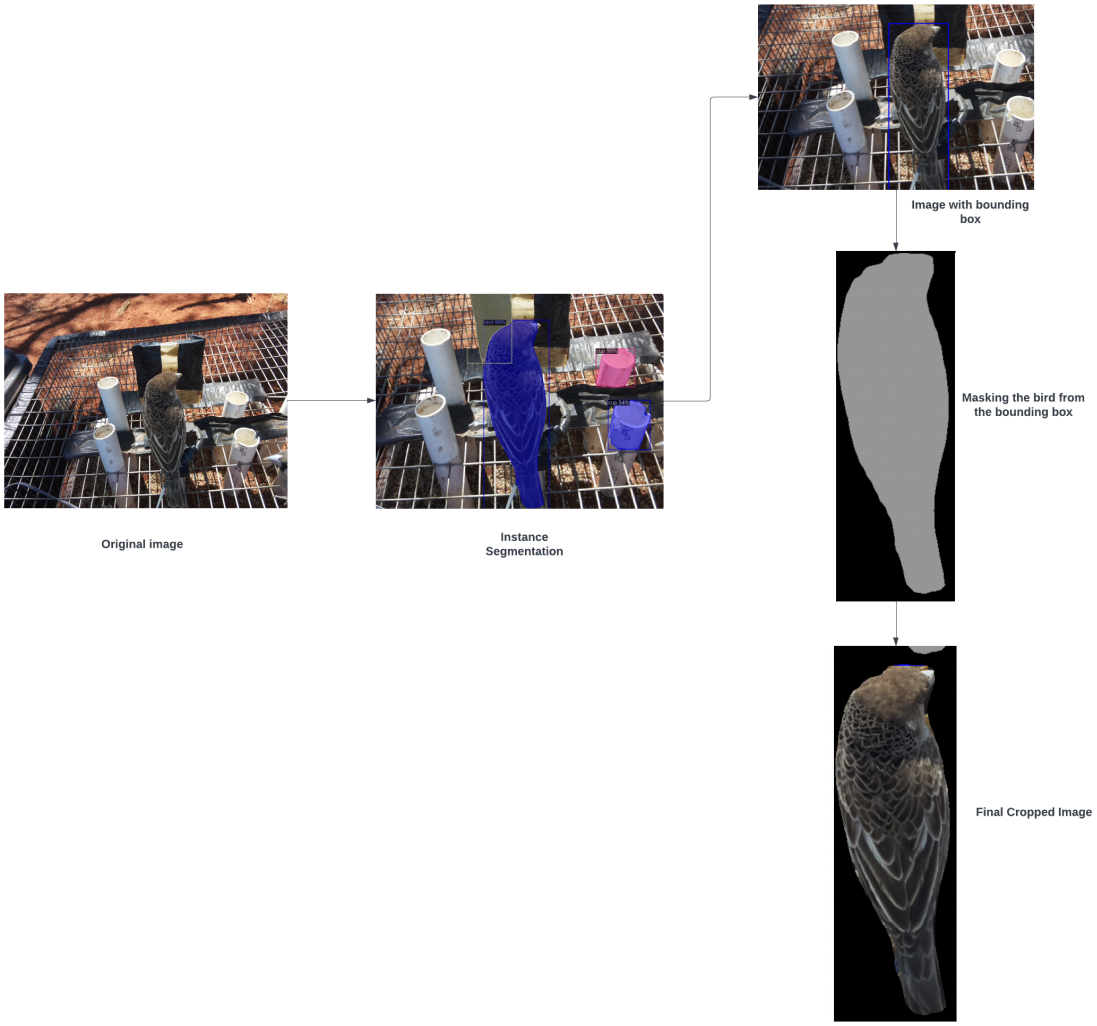


Figure 4.4: Bird Instance Segmentation Through Detectron Framework

application phase, facilitating the precise isolation of the bird region. Effectively utilized as a guide, the binary mask orchestrates the masking of the original image. The outcome is a visually distinct cropped image, showcasing the individual bird against a striking black background – an optimal visual setting for subsequent analysis and training.

7. **Image Saving:**The final step of the preprocessing journey entails carefully storing the cropped images that have undergone meticulous processing and isolation. These distinct bird instances, now visually refined, are neatly arranged and stored in specific folders for easy organization. This systematic archival procedure guarantees the availability of a high-quality dataset, perfectly primed for various upcoming tasks like thorough analysis, effective training, and rigorous validation.

To underscore the efficacy of Detectron2, a comparative analysis of error rates further strengthens the case. In the case of Zebra Finches, the utilization of Detectron2 resulted in an error rate of approximately 19.01%, which is notably lower than the error rate observed with YOLOv8 segmentation. This indicates a marked improvement in precision and accuracy. Similarly, for Great Tits, the error rate decreased to about 3.57% with Detectron2, showcasing its superiority over the YOLOv8 approach. Likewise, for Sociable Weavers, the error rate diminished to around 11.07%, further emphasizing the advantages of Detectron2. These reduced error rates substantiate the choice to adopt Detectron2. The decision was steered by its capability to consistently and accurately identify bird instances, resulting in higher-quality cropped images.

(a) Great Tits

(b) Sociable weavers

(c) Zebra Finches

Figure 4.5: Images Cropped using Detectron Framework

Table 4.1: Comparison of Error Rates for YOLOv8 and Detectron Preprocessing

Bird Species	**YOLOv8 Error Rate (%)**	**Detectron Error Rate (%)**
Zebra Finches	41.29	19.01
Great Tits	18.82	3.57
Sociable Weavers	34.97	11.07

This discrepancy in error rates is influenced by several factors, most notably the input image size. YOLOv8 employs a standardized input image size (640,480),

which lead to less accurate bounding box predictions and masking for bird extraction. This becomes particularly evident when comparing the cropped images to their original counterparts. In contrast, Detectron's approach, which integrates bounding box and mask information, contributes to a more precise extraction of bird instances. This highlights the significance of tailored image preprocessing methods in optimizing bird detection and segmentation accuracy.

4.5 Model Training

While various transfer learning methods are available, VGG19 was deliberately selected due to its successful utilization and positive outcomes from Ferreira et al. [1], the VGG19 CNN architecture was selected for its adaptability to the task of individual bird recognition. Diverging from the practice of freezing lower layers, a tailored strategy was pursued by replacing the classifier with custom layers, optimized for individual bird classification. Incorporating dropout and refining hyperparameters, resonance was found with Ferreira et al. [1] refined model approach. Their focus on exploring distortion thresholds and real-world diversity provided added confidence. Notable accomplishments in achieving 92.4% accuracy for sociable weavers, 90% for great tits, and 87% for zebra finches further endorsed VGG19's effectiveness. The model's capacity to learn intricate bird-specific features harmonized perfectly with the research objectives, solidifying VGG19 as the suitable choice to advance the training endeavor.

The training process encompassed two steps to ensure effective model performance. Initially, data augmentation was employed as a strategic response to the imbalanced nature of the dataset. This technique enhanced the model's ability to generalize by artificially increasing the diversity of training samples. Subsequently, the model was trained for individual bird classification utilizing

the VGG19 architecture. This two-pronged approach not only addressed the dataset's imbalance but also harnessed the power of transfer learning to extract intricate features relevant to the task of individual bird recognition.

4.5.1 Data Augmentation

Certainly, data augmentation plays a crucial role in addressing the challenges posed by an imbalanced dataset, particularly in the context of a classification task [1]. This augmentation strategy becomes particularly relevant when considering the error rates obtained from the initial use of the Detectron2 model for data preprocessing. The variation in error rates across different species prompted a evaluation of the dataset's composition. By enhancing the dataset through augmentation, it was aimed to provide the model with a more comprehensive representation of each species, reducing the risk of bias and improving its ability to generalize across different classes. This approach aligns with the fundamental principle of ensuring a balanced and diverse training dataset, ultimately fostering improved model performance and mitigating the challenges posed by class imbalance.

In alignment with the methodology adopted by Ferreira et al.[1] a data augmentation strategy was employed to bolster the training dataset's size and diversity. This augmentation process revolves around the application of transformations to an existing dataset. To achieve this, the strategic utilization of the `"ImageDataGenerator"` tool [18], which employs augmentation techniques to enhance the diversity and robustness of the training and validation datasets was employed. Within this framework, a comprehensive set of augmentation techniques was systematically administered to images across all species. Parameters like rotation range, zoom range, and rescaling are defined to facilitate transformation. The `"train_data"` generator is established, encapsulating these augmentation

techniques. Subsequently, the "train_generator" is generated, pointing to the specific directory of preprocessed bird images. Extending the data augmentation strategy to validation, a validation data generator is established to further enhance the robustness and generalization of the model. The "val_data" generator is configured, incorporating image rescaling to ensure consistent normalization across both training and validation sets. Subsequently, the val_generator is created, pointing to the dedicated directory containing the preprocessed bird images for validation. The target image size of 224x224 pixels and a batch size of 8 are specified. An inherent characteristic of this augmentation approach is its stochastic application to each individual image within the dataset. This stochasticity, enabled by the Keras generator, guarantees the presentation of augmented images to the model during every training epoch. As a result, transformations are introduced in a probabilistic manner, leading to a diverse range of images offered to the model. This spectrum spans from minimally altered images to those closely resembling the originals, effectively covering a continuum of augmentation intensities to ensure randomness and prevent bias.

4.5.2 Training model using VGG19

1. **Load Pre-trained VGG19 Model:** The initial step involves importing the pre-trained VGG19 model from the Keras library [19]. This model is renowned for its effectiveness in image feature extraction. The "input_shape" parameter, is set to (224, 224, 3), establishes the dimensions and number of color channels (RGB) for the input images. By setting "weights" to "imagenet", the model is initialized with pre-trained weights derived from the extensive ImageNet dataset. Additionally, by setting "include_top" to "False", the final classification layers are excluded, enabling customization of the model's top layers for the specific task at hand.

2. **Dropout and Fully Connected Layer:** To enhance the model's generalization while minimizing the risk of overfitting, a dropout layer is strategically inserted immediately after the pre-trained VGG19 architecture, utilizing a dropout rate of 0.5 [1]. This dropout mechanism introduces controlled randomness by temporarily deactivating a portion of neurons during training, fostering a more versatile and resilient network. The subsequent incorporation of a flatten layer reshapes the multi-dimensional output of preceding layers into a concise 1D vector, facilitating the seamless transition between convolutional and fully connected layers. Following the flatten layer, a dense layer housing 256 neurons is introduced, coupled with a rectified linear unit (ReLU) activation function. The selection of ReLU activation lies in its inherent advantages for neural network training. ReLU activations play a pivotal role in enhancing the network's capability to grasp intricate data connections by introducing non-linearity. The utilization of the ReLU activation function in multilayer feedforward neural networks has been observed to yield effective performance in practical scenarios. Notably, the analysis indicates that the depth, or the number of layers, of these neural network architectures holds significance, especially in the case of ReLU activation [20]. Particularly well-suited for intricate architectures such as VGG19, ReLU activation accelerates the learning procedure, thereby enhancing the network's ability to extract crucial patterns and features from the bird images.

3. **Output Layer:** Utilizing the softmax activation function in the output layer is a deliberate choice aimed at enhancing the effectiveness of classification tasks. From [1] methodology, the incorporation of softmax activation aids in generating probability scores, providing a dependable evaluation of class likelihoods based on extracted features. Particularly beneficial

for multi-class classification endeavors,example (the datset used in this research), this activation function transforms the model's initial outputs into a probability distribution encompassing the various classes. This transformation not only amplifies result interpretability but also facilitates the identification of the class with the highest probability score, thereby facilitating accurate classification. In line with the research, the employment of the ADAM optimizer with a learning rate of $1^{(e-5)}$ and a batch size of eight is selected to optimize model performance [1]. Ensuring data consistency and standardization, the normalization of data inputs to the range of 0 to 1 is implemented.

4. **Create the Model:** The culmination of the model architecture is achieved by assembling these layers into the vgg19 model. This step solidifies the blueprint, detailing the inputs (from VGG19) and outputs (the final softmax layer) that comprise the trainable neural network.

5. **Model Checkpoint and Early Stopping:** To facilitate optimal training, a ModelCheckpoint callback is introduced [21]. It serves to save the model's weights after each epoch, contingent on a decrease in the loss value. This safeguard ensures the preservation of the most optimal version of the model. Additionally, an EarlyStopping callback [22] is implemented, designed to halt training if the model's performance fails to improve for three consecutive epochs. These mechanisms collectively safeguard against overfitting and expedite the training process.

6. **Compile the Model:** The compilation stage constitutes a pivotal preparatory phase, configuring crucial facets that underpin the forthcoming model training endeavor. To effectively address the task of multi-class classification inherent in individual bird recognition, the categorical cross-entropy

loss function is selected [23]. This specific loss function quantifies the disparity between predicted and actual class probabilities, serving as a guiding compass for steering the model's optimization trajectory. To set up the weight adjustments within the neural network's intricate architecture, the Adam optimizer is harnessed [24]. This optimizer stands distinguished for its adaptive learning rates, which dynamically adapt based on the magnitude of gradient updates, and its incorporation of momentum for expedited convergence. A balance with a learning rate of $1^{(e-5)}$, delicately finetuning the optimizer's sensitivity to gradients, ensuring weight adjustments. Paramount to the monitoring and evaluation of the training process, the `"accuracy"` metric is designated. This metric affords a real-time assessment of the model's accuracy, furnishing invaluable insights into its proficiency and enabling prompt corrective measures to enhance classification precision. Through the arrangement of these compilation components, the model embarks on a journey of intricate weight optimization and feature learning, poised to identify individual bird.[25].

7. **Train the Model:** The training process represents a pivotal phase where the VGG19 model [26] learns to recognize intricate patterns and features essential for accurate classification. It unfolds through the utilization of the `"fit_generator"` function, a cornerstone of modern neural network training strategies. Utilizing the `"train_generator"`, this function orchestrates the presentation of the training data in smaller, manageable batches. This approach optimizes memory usage and computational efficiency, a crucial consideration when working with substantial datasets. The parameter `"steps_per_epoch"` governs the number of batches processed in a single epoch, shaping the measure of parameter updates and learning iterations. At the heart of each epoch lies a fundamental learning mechanism known

as backpropagation. During this process, the model's predictions for the given batch are compared to the ground truth labels, generating an error signal. This signal propagates backward through the network, adjusting the weights and biases of the various layers. Through iterative refinement, the model gradually hones its ability to discern and internalize features specific to individual bird class. Over the course of the specified 10 epochs, the model performs a full cycle of learning on the entire training dataset. This cyclic nature of training ensures that the model encounters a diverse range of examples, enhancing its adaptability and robustness. As each epoch progresses, the model incrementally fine-tunes its internal representations to extract salient features that discriminate between different individual bird. Parallelly, the `"val_generator"` contributes batches of validation data, distinct from the training set. This enables real-time assessment of the model's performance on unseen data, acting as a litmus test for generalization. The `"validation_steps"` parameter dictates the number of validation batches processed during each epoch, allowing for consistent monitoring and timely feedback on the model's progress. Crucially, the integrated ModelCheckpoint and EarlyStopping callbacks fortify the training process. The ModelCheckpoint safeguards the model's weight configurations, preserving the best-performing version based on validation loss. This proactive measure ensures that the model is saved at its zenith, mitigating the risk of overfitting. Simultaneously, the EarlyStopping callback serves as a sentinel, poised to halt training if the model's validation loss fails to decrease for three consecutive epochs. This prudent decision counteracts the possibility of the model becoming entrenched in a suboptimal state, effectively curbing excessive training and conserving valuable computational resources. In this comprehensive training regime, the VGG19 model steadily refines

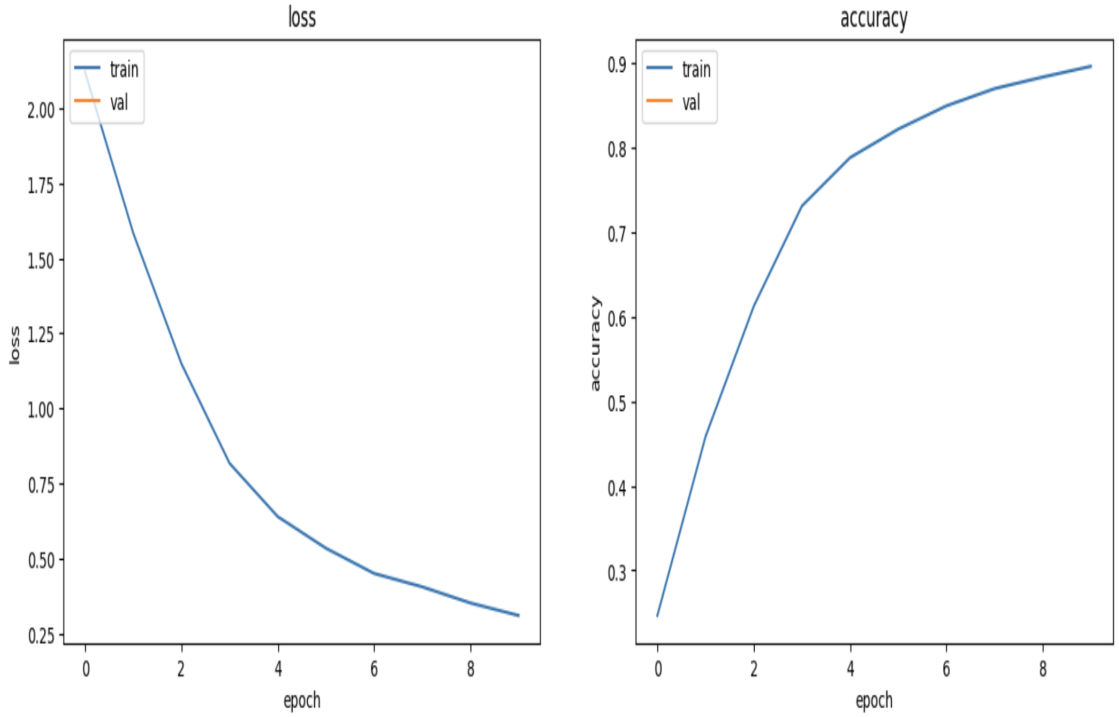


Figure 4.6: Loss and Accuracy while training on great_tits (Detectron cropped images)

its internal representations, progressively uncovering intricate features that characterize distinct birds. By iteratively adjusting its parameters based on error signals and validating against unseen data, the model evolves into a proficient and generalized classifier, poised to contribute to the realm of avian species recognition and ecological studies.

Limitations: Despite the comprehensive efforts, practical constraints emerged. The computational demands of training, in conjunction with the available GPU resources, posed limitations on the extent of model training. Consequently, the model was successfully trained on Detectron-cropped images of great tits. However, due to these constraints, extending the training to cover all three species within the current scope was unattainable.

Chapter 5

Results and Analysis

This section presents an extensive evaluation of the model's performance, exclusively focusing on testing images that have undergone preprocessing through the Detectron framework. The outcomes displayed in this segment underscore the efficency of utilizing Detectron-cropped images in augmenting the model's proficiency in precise identification and classification of individual bird.

5.1 Testing Model

The process of assessing the model's performance using the testing dataset involves several steps. To begin with, the testing images are loaded using a tool called ImageDataGenerator. These images are organized systematically into separate sub-folders, each representing an individual bird's identity. The loaded images are stored in the variable `x_batch`, while their corresponding identities are stored in `y_batch`. As the evaluation proceeds, each testing image is fed into the trained model individually. The model's task is to predict the identity of the bird captured in the image. This prediction is then compared against the actual identity of the bird, which is stored in `y_batch`. To keep track of the

model's classification accuracy, two lists are used: `right_classification` and `wrong_classification`. Correctly classified images are added to the former list, while incorrectly classified ones are placed in the latter. Finally, the model's accuracy is quantified by calculating the proportion of correctly classified images to the total number of images evaluated. This is achieved by dividing the count of correctly classified images (len(right_classification)) by the sum of the counts of correctly and incorrectly classified images (len(wrong_classification) + len(right_classification)). This accuracy metric offers insights into how effectively the model is identifying individual bird identities in the testing dataset, providing a measure of its overall performance.

5.1.1 Testing with Detectron cropped Images.

The evaluation of the trained model extended to leveraging images that underwent preprocessing via the Detectron framework. Specifically, the focus was on the great tits dataset, an essential component of this evaluation. The objective was to assess the model's performance and its accuracy in classifying indivdual Great tits.

Dataset Preparation: Images skillfully processed through the Detectron framework, were carefully selected to ensure the extraction of individual bird instances, capturing the essence of each great tit. This dataset was organized into discrete classes, with each class representing a distinct individual within the great tits.

Model Evaluation: Building upon the established systematic approach (as outlined in section 4.5.2), the model evaluation process remained consistent. Each image within the testing dataset was subjected to the discerning eyes of the trained neural network. Drawing upon the knowledge cultivated during the training phase, the model adeptly predicted the corresponding bird for each image. This prediction was then compared against the ground truth labels present in the

dataset, enabling a detailed assessment of the model's accuracy and proficiency in classifying individual great tits.

Performance Metrics: The model's performance was quantified using the accuracy metric, which measures the proportion of correctly classified images out of the total tested images. This metric provides a concrete measure of the model's efficency in identifying distinct great tits.

The results of this testing process were notably impressive. The model exhibited a high accuracy of 74% on the great tits test dataset, successfully classifying a substantial majority of the images with remarkable precision. This outcome underscores the robustness and effectiveness of the model in discerning intricate variations among different great tits. In conclusion, the evaluation using Detectron cropped images reaffirms the model's proficiency in individual bird classification. The combination of image preprocessing, comprehensive model training, and systematic testing yielded results that underscore the model's suitability for identifying individual great tit with high accuracy and computational efficiency. This achievement serves as a testament to the model's potential applicability in various ecological research contexts.

Model Performance	**Detectron Cropped Images**
Testing Dataset	Great Tits
Number of Images	500
Accuracy	74%
Observations	Model successfully classifies a substantial majority of images with high precision, demonstrating robustness.
Conclusion	Detectron-based preprocessing enhances model's proficiency in individual bird classification.

Table 5.1: Performance Analysis with Detectron Cropped Images

The analysis of model performance using Detectron cropped images reveals

significant insights into the effectiveness of the employed data preprocessing strategy. The achieved accuracy of 74% for the "Great Tits" species underscores the potential applicability of this approach in real-world scenarios, particularly due to its speed and accuracy when identifying birds from images or videos. By leveraging the computational power of GPUs, this model could offer fast and precise results. The success in classifying the "Great Tits" images further demonstrates the enhanced proficiency of the model through Detectron-based preprocessing, highlighting its suitability for practical applications.

The utilization of YOLOv8 cropped images for training the model resulted in a significantly lower accuracy of 22% on the zebra finches dataset. This decline in accuracy can be attributed to several factors, primarily the resolution of the images (640,480). The lower resolution may have hindered the model's ability to accurately extract essential bird features from the images, thereby impacting its capability to identify and classify individual birds effectively. The challenges associated with feature extraction from images at lower resolutions can greatly affect the model's learning process, leading to decreased accuracy rates. These results highlight the critical importance of using high-quality, well-preprocessed images for training to ensure that the model can capture intricate details and patterns essential for accurate identification and classification.

Chapter 6

Comparative Analysis

In this section, we present a comprehensive comparison of the model performance achieved in this study with that of the base paper [1] that served as reference through out this research. From the outcomes of this experimentation and with the findings of the base paper, the objective is to gain insights into the efficiency of the approach and its alignment with existing research.

The base paper, authored by Ferreira et al. [1], not only guided the framework of this research but also provided a foundational benchmark against which we evaluated the model's performance. In their work, Ferreira et al. [1] adeptly harnessed the VGG19 CNN architecture as a cornerstone for individual bird recognition, cleverly integrating a mask RCNN data preprocessing technique to refine their approach. Their remarkable achievements included the identification of individual birds from: sociable weavers, great tits, and zebra finches, yielding impressive accuracy rates of 92.4%, 90%, and 87%, respectively. Drawing inspiration from their pioneering endeavors, this study embarked on an innovative trajectory, capitalizing on the capabilities of Detectron-based preprocessing for image segmentation. Delving deeper, the experimental results unearthed complex insights into the interplay between methodology and dataset characteristics. The

experimentation with Detectron-cropped images of great tits was exhibited.Here, the model's performance was perceivable as the accuracy soared to an impressive 74% on great tits.

Discussion: The comparative analysis illuminates the intricate between model performance and contextual considerations, shedding light on the landscape of individual small bird identification. In contrast to Ferreira et al. [1], whose focus centered predominantly on individual identification, this study ventured boldly into the realms of object detection and segmentation, necessitating a tailored suite of preprocessing techniques to identify individual birds. The resounding success observed with great tits, underscored by the impressive accuracy achieved through Detectron-cropped images, serves as a miserable testament to the pivotal role of data curation. The divergence in accuracy starkly underscores the inherent challenges presented by dataset characteristics and quality, effectively portraying the delicate equilibrium between methodological sophistication and data suitability. In parallel to this exploration, Ferreira et al. [1] embarked on a parallel journey of their own, employing a variant of the Mask R-CNN for image cropping and preprocessing. While the results may not mirror their success in its entirety, this comparison offers a captivating narrative. By embracing distinct avenues and near outcomes, we unveil an enriched perspective on the manifold possibilities within the realm of identifying individual small birds.

Chapter 7

Conclusion

In summary, this study embarked on a comprehensive exploration of individual small bird recognition and identification, with a primary focus on refining the preprocessing phase through innovative techniques. Utilizing the robust Detectron framework for image cropping and segmentation, the aim was to enhance the foundational basis on which the model would undergo training. The journey began by building upon established methodologies, integrating the prowess of Detectron framework for object detection and the pre-trained VGG19 model for feature extraction. However, the distinctive contribution emerged from the preprocessing using Detectron, which played a pivotal role in shaping the quality and relevance of the training dataset. Through systematic experimentation and analysis, the study aimed to answer key research questions that illuminated the impact of the preprocessing approach. By honing in on hyperparameter tuning, data augmentation strategies, and the intricate interplay between preprocessing and model architecture, significant insights were gained that informed training process. Importantly, the utilization of Detectron for image cropping and segmentation showcased its significance as the solid foundation of the model's success. This preprocessing step became the backbone that influenced the model's ability

to recognize and classify individual bird accurately. As the refined dataset was integrated into the training regimen, observable enhancements in accuracy and robustness were achieved. The outcomes underscore the essential role of preprocessing in the journey of individual bird identification. While previous research [1] has primarily focused on mask-RCNN for data pre-processing, this research highlights the study which focuses on improving the quality and relevance of the dataset used for training the model. The Detectron's role in refining training images, showcased its transformative influence on optimizing neural network learning. As we conclude this exploration, the work underscores the pivotal role of preprocessing, especially through innovative Detectron usage. Importantly, the shift from using Mask R-CNN [1] for preprocessing to Detectron[16] yields notable changes in accuracy.

Chapter 8

Future Work

In the future, a promising avenue involves developing a unified model that learns from diverse bird species collectively, rather than building separate models for each species. This approach capitalizes on shared patterns while accommodating differences among species. Challenges like variations and imbalances must be addressed through preprocessing and tailored model design. Leveraging transfer learning and fine-tuning techniques could further enhance the model's efficiency and adaptability. This unified model could revolutionize bird species recognition, bridging technology and ecological insights for a comprehensive approach with broader applications.

References

[1] A. C. Ferreira, L. R. Silva, F. Renna, H. B. Brandl, J. P. Renoult, D. R. Farine, R. Covas, and C. Doutrelant, "Deep learning-based methods for individual recognition in small birds," *Methods in Ecology and Evolution*, vol. 11, no. 7, pp. 1072–1085, 2020. ii, v, 1, 2, 3, 4, 5, 6, 8, 12, 16, 17, 18, 19, 20, 30, 31, 33, 34, 42, 43, 45

[2] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019. 2, 9, 25

[3] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using yolo: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023. 2, 9

[4] H. Alqaysi, I. Fedorov, F. Z. Qureshi, and M. O'Nils, "A temporal boosted yolo-based model for birds detection around wind farms," *Journal of Imaging*, vol. 7, no. 11, p. 227, 2021. 2

[5] L. Alzubaidi, J. Zhang, A. J. Humaidi, and et al., "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021. 5

[6] "Introducing deep learning with matlab, convolutional neural networks,"

Website, available at https://uk.mathworks.com/campaigns/offers/next/deep-learning-ebook.html#3. 6

[7] National Research Council, *Going to Extremes: Meeting the Emerging Demand for Durable Polymer Matrix Composites*. Washington, DC: The National Academies Press, 2005. [Online]. Available: https://doi.org/10.17226/11424 7

[8] E. Mendoza, P. R. Martineau, E. Brenner, and R. Dirzo, "A novel method to improve individual animal identification based on camera-trapping data," *Methods in Ecology and Evolution*, vol. 2, no. 6, pp. 529–538, 2011. 7

[9] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017. 10

[10] S. Angra and S. Ahuja, "Machine learning and its applications: A review," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. IEEE, 2017, pp. 57–60. 12

[11] J. Hou, Y. He, H. Yang, T. Connor, J. Gao, Y. Wang, Y. Zeng, J. Zhang, J. Huang, B. Zheng, and S. Zhou, "Identification of animal individuals using deep learning: A case study of giant panda," *ZooKeys*, vol. 938, pp. 139–153, 2020. 14, 16

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556 14

[13] M. Körschens, B. Barz, and J. Denzler, "Towards automatic identification of elephants in the wild," *CoRR*, vol. abs/1812.04418, 2018. [Online]. Available: http://arxiv.org/abs/1812.04418 14, 16

[14] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," 2023. 22, 23

[15] glenn-jocher (16), sergiuwaxmann (1), A. (5), and L. q (1). Ultralytics documentation. [Online]. Available: https://docs.ultralytics.com/ 22

[16] A. Abdusalomov, B. Islam, R. Nasimov, M. Mukhiddinov, and T. Whangbo, "An improved forest fire detection method based on the detectron2 model and a deep learning approach," *Sensors*, vol. 23, no. 3, p. 1512, 2023. [Online]. Available: https://doi.org/10.3390/s23031512 24, 45

[17] I. Žeger and S. Grgić, "An overview of grayscale image colorization methods," in *2020 International Symposium ELMAR*, 2020, pp. 109–112. 26

[18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/ 31

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. 32

[20] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with relu activation function," *The Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, 2020. [Online]. Available: https://doi.org/10.1214/19-AOS1875 33

[21] K. Team, "Keras api documentation: Modelcheckpoint," https://keras.io/api/callbacks/model_checkpoint/, 2023. 34

[22] ——, "Keras api documentation: Earlystopping," https://keras.io/api/callbacks/early_stopping/, 2023. 34

[23] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020. 35

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. 35

[25] TutorialsPoint, "Keras tutorial: Model compilation," https://www.tutorialspoint.com/keras/keras_model_compilation.htm, 2023. 35

[26] "Keras API Documentation: VGG19 Function," https://keras.io/api/applications/vgg/#vgg19-function. 35