**Title**
Deep learning applications in wildlife recognition

**Author**
Miao, Zhongqi

**Publication Date**
2022

Deep learning applications in wildlife recognition

by

Zhongqi Miao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Environmental Science, Policy, and Management

and the Designated Emphasis

in

Computer and Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Wayne M. Getz, Co-chair
Professor Stella X. Yu, Co-chair
Professor Carl Boettiger
Professor Dorit S. Hochbaum
Professor William D. Collins

Spring 2022

Deep learning applications in wildlife recognition

# Abstract

Deep learning applications in wildlife recognition

by

Zhongqi Miao

Doctor of Philosophy in Environmental Science, Policy, and Management

and the Designated Emphasis in

Computer and Data Science and Engineering

University of California, Berkeley

Professor Wayne M. Getz, Co-chair

Professor Stella X. Yu, Co-chair

Deep learning has attracted much attention from the ecological community for its capability of extracting and generalizing patterns from data sets with highly complicated structures, such as images, audios, and motion signals. However, despite the promising cases, deep learning is complicated in terms of application and has shortcomings when applied to real-world ecological data sets. In this dissertation, we focus on: 1) demystifying the hidden mechanisms of deep learning in terms of wildlife recognition, 2) identifying the challenges of deep learning applications in wildlife recognition, and 3) proposing a generic recognition framework that can be practically deployed in the fields.

In the first chapter, we examine how deep learning recognizes wildlife through Convolutional Neural Network feature deconstruction and interpretation. The objective is to demystify aspects of artificial intelligence and facilitate wildlife recognition research.

The second chapter identifies three major challenges to automatic wildlife recognition through an avian recognition case study and provides preliminary solutions addressing each challenge. This chapter aims to increase awareness in the ecological community of these challenges, bridge the gap between ecological applications and state-of-the-art computer science, and open doors to future research.

In the third chapter, we propose a hybrid recognition system of machine learning and human-in-the-loop that overcomes two challenges discussed in the second chapter: imbalanced data distribution and continuous data expansion. Moreover, with the self-updating mechanism of our approach, the system can be practically deployed in the fields.

# Contents

# Acknowledgments

First, I want to thank my advisors, Professor Wayne Getz and Professor Stella Yu. Without the help from both of them, I wouldn't be able to get this far. They accepted me into their lab when I had almost no achievements, provided me complete independence to explore uncharted areas, and offered me invaluable guidance that led to this dissertation. I was fortunate to learn from the best in wildlife ecology and machine learning and be able to work on problems in both areas at the same time.

I want to thank my committee members Professor Carl Boettiger, Professor Dorit Hochbaum, and Professor William Collins for their precious time and effort.

Thanks to Professor Ziwei Liu, who has provided me with significant guidance in machine learning and allowed me to participate in his computer vision projects. To Professor Kaitlyn Gaynor, who has provided her hard collected data set collected from Africa. It was her data set that made most of my projects possible.

Thanks to members from USGS, Luke Fara, Enrika Hlavacek, Travis Harrison, Mark Koneff, Kyle Landolt, Bradley Pickens, and Timothy White for supporting me for two semesters and providing hard work in the avian recognition project.

Thanks to Whitney Mgbara, Ludovica Vissat, Nir Horvitz, Oliver Muellerklien, Eric Dougherty, Dana Seidel, and Colin Carlson from the Getz Lab and Tsung-Wei Ke, Jyh-Jing Hwang, Frank Wang, Peter Wang, Zhirong Wu, Bala Yellapragada, and Sacha Hornauer from the Yu's Lab. Thanks for being great friends, inspiring co-workers, and insightful collaborators.

Thanks to members from the Kitzes Lab, Professor Justin Kitzes, Sam Lapp, and Tessa Rhinehart, for introducing me to the world of bird audios.

Thanks to all my past collaborators from all over the departments, schools, and the world, Meredith Palmer, Mohammad Norouzzadeh, Alex McInturff, Rauri Bowie, Ran Nathan, Xiaohang Zhan, Xingang Pan, Boqing Gong, Dahua Lin, for providing hard effort in our collaborations.

Finally, thanks to my family and friends for being supportive for the past couple of years.

# Introduction

## Background

The world is undergoing rapid global climate change and loss of biodiversity [42, 6, 104, 31]. Capturing responses of wildlife to environmental and anthropogenic perturbations has become crucial to understanding long-term patterns, drivers, and consequences of species declines and extinctions [121, 102]. Imagery-based, long-term, and large-scale remote sensing with devices such as motion-sensitive cameras (camera traps) [77, 97] and aerial devices (e.g., aircraft and satellites) [142, 80] is one of the most widely adopted, non-invasive, and cost-efficient methods for wildlife monitoring. However, the most significant drawback of this method is the amount of human labor needed to annotate wildlife images for ecological analysis in order to keep pace with continuous data collection at a management-relevant timescale [101, 3, 124, 86, 85].

Artificial Intelligence (AI), particularly deep learning algorithms that employ convolutional neural networks (CNNs), is the breakthrough technology of the current half-decade and provides promising solutions for rapid and high-accuracy image recognition, including wildlife recognition [65, 86, 127, 145]. Therefore, it may significantly increase the efficiency of associated ecological studies [137, 85]. In the interest of ecologists being able to apply deep learning methods more effectively and more efficiently to their particular fields of investigation, in this dissertation, we focus on:

1. Demystifying the hidden mechanisms of deep learning in terms of wildlife recognition.
2. Identifying the challenges of deep learning applications in wildlife recognition.
3. Proposing a generic recognition framework that addresses the challenges and can be practically deployed in the fields.

## Related works

Automatic wildlife image recognition has been attempted in various directions in the past [125, 33, 149]. Before the deep learning era, most methods rely on hand-crafted image features for animal detection and species classification tasks. For example, Swinnen et al. [125] and Figueroa et al. [33] used similar approaches by calculating average pixel changes in images to detect the existence of animals. However, simple animal detection cannot fulfill the requirements for practical wildlife applications because it cannot provide detailed species-level information. Yu et al. [149], on the other hand, successfully applied conventional computer vision methods such as Scale-Invariant Feature Transform (SIFT) [76] and Histogram of Oriented Gradients (HOG) [24] to convert wildlife images into linearly classifiable features. Despite the promising results ($\sim$ 80% species-level test accuracy), hand-crafted visual features like SIFT and HOG usually require clear images of animals with distinctive body marks (e.g., black spots of leopards) to produce high-quality features for classification [149]. However, automatically collected wildlife images usually do not have clear views of animals, limiting the quality of hand-crafted features. Moreover, hand-crafted features require independent parameter tuning for every category to produce optimal results, which is time-consuming and relies on domain expertise [109, 149].

Deep learning methods have proven to have good generalization abilities without relying on hand-crafted features [65]. It can automatically learn linearly classifiable features through a large quantity of training data. Chen et al. [18] and Gomez et al. [37] are the two earliest studies that applied deep learning techniques to classify wildlife from real-world camera trap images with limited image quality. However, the model from Chen et al. [18] did not perform well at species-level classification ($\sim$ 40% test accuracy) because of the limitations of shallow networks they used in their experiments and the lack of training data (only 20,000 training images). On the other hand, Gomez et al. [37] only reported experiments of binary classifications between birds and mammals, which did not provide detailed species-level information.

Gomez et al. [137] and Norouzzade et al. [85] are the first two studies that had successful results of deep learning applications in wildlife recognition from large-scale camera trap data sets. Gomez et al. achieved $\sim$ 80% test accuracy on a 26-class wildlife data set, and Norouzzade et al. achieved over 90% test accuracy on a 48-class wildlife data set. Norouzzade et al. also proposed to use deep learning to coarsely classify animals behaviors and count the number of individuals in images that contain multiple animals [85].

After these two successful approaches, deep learning has been extensively discussed on possible applications to all sorts of ecological data with complicated structures, such as overhead images [50, 72], bio-acoustics [54, 1], earthquake signals [112], thermal signals [21] and time-frequency signals [16, 52]. However, despite the success in experiments of previous studies, it is still rare to see deep learning being deployed in the fields and providing reliable automatic recognition results that can be used for further downstream tasks such as wildlife population modeling. We identify two major reasons that are hampering the deployment of deep learning methods in practice:

1. The lack of background knowledge in machine learning and deep learning makes it difficult for the ecological community to understand the mechanisms of deep learning methods in wildlife recognition. Therefore, it is hard for people in the fields to figure out the reasons for failures (i.e., misclassifications) and make meaningful adjustments to previous models to adapt to specific requirements of different studies and data sets.

2. Most of the previous deep learning studies mainly focus on reporting the performance of their approaches while ignoring the challenges from real-world data sets that can limit the model performance. For example, in Norouzzade et al. [85], the data set in their experiments were artificially balanced. Likewise, in Gomez et al. [137], the authors excluded classes with limited training images to preserve the model performance. This

is especially true in the computer vision community, where most of the studies only focus on the performance of standard benchmarks (e.g., ImageNet [27] and Microsoft-Common Object data set [70]), which usually have artificial properties like balanced class distributions and high image quality. However, the challenges in real-world data, such as imbalanced distribution, low image quality, and long-term data expansion, are often the limiting factors of the deployment of deep learning models and require extensive discussion.

In contrast to previous studies, in this dissertation, we focus on three questions that lack enough discussion:

1. How can we interpret the hidden mechanisms behind deep learning, particularly in wildlife recognition?
2. What are the challenges that limit the performance of deep learning in practice?
3. Considering the practical challenges, how can we build a deployable wildlife recognition system in the fields?

## Summary of chapters

In the first chapter, we explain how deep learning models recognize wildlife from imagery. First, we deconstructed the features used by a CNN that we trained to identify 20 different species in more than 100,000 annotated wildlife images obtained from a national wildlife park in Mozambique. Through this deconstruction, we show that the features used by AI models in wildlife recognition were similar to, but in some cases, different from those used by humans. We also provide visualizations of feature space structures in the form of a visual similarity tree that presents a clear view of how the AI model in our experiment perceived similarities and differences among species.

After discussing the mechanisms of deep learning, in the second chapter, we identify three major challenges faced in automatic wildlife recognition: 1) extremely imbalanced data distribution, 2) annotation uncertainty in categorization, and 3) continuous data expansion. We also present solutions addressing these challenges through a case study of avian recognition with data collected from two over-water study sites, the Atlantic Ocean near Cape Cod, MA and Lake Michigan, MI. Our objective is to demystify the affecting factors of AI recognition performance and demonstrate the flexibility of deep learning methods.

Finally, in the third chapter, we propose an iterative human and automated recognition framework that makes AI systems deployable with efficient human-in-the-loop and imperfect classification models. We address two challenges discussed in the second chapter with our dynamic recognition framework: imbalanced data distribution and continuous data expansion. As a case study, this framework was applied to a long-term, large-scale camera trap project from Gorongosa National Park, Mozambique. Because of the necessity of human annotation for novel species, the goal of our framework is to minimize the need for human intervention by applying human annotation solely on difficult images or novel species while maximizing the recognition performance of each model update procedure when new data are collected.

With a synergistic collaboration between humans and machines, the role of deep learning is transformed from a remote tool that does all the work to an adaptive tool that vastly relieves the burden of human annotation. In return, the incorporation of human-in-the-loop enables efficient and constant model updates.

# Chapter 1

# Insights and approaches using deep learning to classify wildlife

Zhongqi Miao[1,2], Kaitlyn M Gaynor[1], Jiayun Wang[2,3], Ziwei Liu[2], Oliver Muellerklein[1], Mohammad Sadegh Norouzzadeh[4], Alex McInturff[1], Rauri C K Bowie[5], Ran Nathan[6], Stella X Yu[2,3], Wayne M Getz[1,7],

This chapter is published in Scientific Report [82] and is included as a dissertation chapter with permission from co-authors.

[1]Dept. Env. Sci., Pol. & Manag., UC Berkeley, CA, United States
[2]International Comp. Sci. Inst., UC Berkeley, 1947 Center St, Berkeley, CA, United States
[3]Vision Sci. Grad. Group, UC Berkeley, CA, United States
[4]Dept. Comp. Sci., U. Wyoming, WY, United States
[5]Dept. Integr. Biol. & Museum of Vertebrate Zoology, UC Berkeley, CA, United States
[6]Dept. EEB, Alexander Silberman Inst. Life Sci., Hebrew U. Jerusalem, Givat Ram, Israel
[7]Sch. Math. Sci. Univ. KwaZulu-Natal, South Africa

# Abstract

The implementation of intelligent software to identify and classify objects and individuals in images is a technology of growing importance to operatives in many fields, including wildlife conservation and management. However, to non-experts, the methods can be abstruse and the results mystifying. Here, in the context of applying cutting-edge methods to classify wildlife species from camera trap data, we shed light on the methods themselves and the types of features these methods extract to make classifications. The current state of the art is to employ deep learning with convolutional neural networks (CNNs). We outline these methods and present results obtained in training a CNN to classify 20 African wildlife species with an overall accuracy of 83.0% from a dataset containing 111,467 images. We demonstrate the application of a class activation mapping procedure to extract the most salient pixels in the final convolution layers. We show that in our experiment, these pixels highlight features in particular images, in some cases, were similar to those used to train humans to identify animal species. Further, we used Mutual Information to identify the neurons in the final convolution layers that consistently responded most strongly across a set of images of one particular species. We then interpret the features in the images where the strongest responses occurred and present dataset biases revealed by these extracted features. We also used hierarchical clustering of feature vectors associated with each image to produce a visual similarity dendrogram of identified species. Finally, we illustrate the relative unfamiliarity of test images (i.e., relative feature distances to the feature centroids of known species) to demonstrate the capacity of CNNs to recognize unknown species.

# Acknowledgements

Thanks to T. Gu, A. Ke, H. Rosen, A. Wu, C. Jurgensen, E. Lai, M. Levy, and E. Silverberg for annotating the images used in this study and to everyone else involved in this project. Data collection was supported by J. Brashares and through grants to KMG from the NSF-GRFP, the Rufford Foundation, Idea Wild, the Explorers Club, and the UC Berkeley Center for African Studies. We are grateful for the support of Gorongosa National Park, especially M. Stalmans, in permitting and facilitating this research. ZM was funded in part by NSF EEID Grant 1617982 to WMG, RCKB, and RN and was also supported in part by BSF Grant 2015904 to RN and WMG. Thanks to Z. Beba, T. Easter, P. Hammond, Z. Melvin, L. Reiswig, and N. Schramm for participating in the feature survey.

# Introduction

Collecting animal imagery data with motion-sensitive cameras (camera traps) is a minimally invasive approach to obtaining relative densities and estimating population trends in animals over time [77, 97]. It enables researchers to study their subjects remotely by counting animals from the collected images [13]. However, images are not readily analyzable in their raw form due to their complexity, and relevant information must be visually extracted. Therefore, human labor is currently the primary means to recognize and count animals in images. This bottleneck impedes the progress of ecological studies that involve image processing. For example, in the Snapshot Serengeti camera trap project, it took years for experts and citizen scientists to manually label millions of images [85].

Deep learning methods [65] have revolutionized our ability to train computers to recognize all kinds of objects from imagery data, including faces [129, 73] and wildlife species [85, 128, 140]. It may significantly increase the efficiency of associated ecological studies [137, 85]. In our quest to demystify the methods, it would be useful to have machines articulate the features they employ to identify objects [94, 17]. This articulation would not only allow machines to communicate more intelligently with humans but may also allow machines to reveal the weakness of the methods, dataset biases, and cues that humans are currently not using for object identification, which could then make humans more effective at such identification tasks. However, we must identify the human-understandable visual features machines use to classify objects before we can do this. To the best of our knowledge, none of the existing studies that use deep learning for animal classification concentrate on this issue. As such, they lack the necessary transparency for effective implementation and reproducibility of deep learning methods in wildlife ecology and conservation biology.

To identify such features in the context of classification of wildlife from camera trap data, we trained a standard Convolutional Neural Network (CNN) [64, 140] using a deep learning algorithm on a fully annotated dataset from Gorongosa National Park, Mozambique, that has not previously been subjected to machine learning. After training, we interrogated our network to understand better the features it used to make classifications by deconstructing the features on the following three aspects of our implementation: 1) localized visual features, 2) common within-species visual features, and 3) interspecific visual similarities (Figure 1.1).

We used Guided Grad-CAM (GG-CAM)—a combination of Guided Back-propagation (GBP) [119] and gradient-weighted class activation mapping (Grad-CAM) [111]—on the last convolutional layer of our trained network to extract localized visual features of single images. We can obtain indirect reasons for the CNN classifications by inspecting the results. In addition, we conducted a relatively informal experiment that compared extracted features with visual descriptors used by human annotators to identify species in our image sets of corresponding animal species. We found that, to some extent, the features used by the CNN to identify animals were similar to those used by humans. Next, we used Mutual Information (MI) [8, 78] to generalize within-species features as an extension of the localized visual features of single images. These generalized features revealed inner biases of the dataset. Then, we used hierarchical clustering [105] on the CNN feature vectors to further inspect the visual similarities between animal species learned by the CNN. Again, we found that the relative visual similarities that emerged during the training process were similar to how humans perceived animal visual similarities. Finally, we measured the relative unfamiliarity

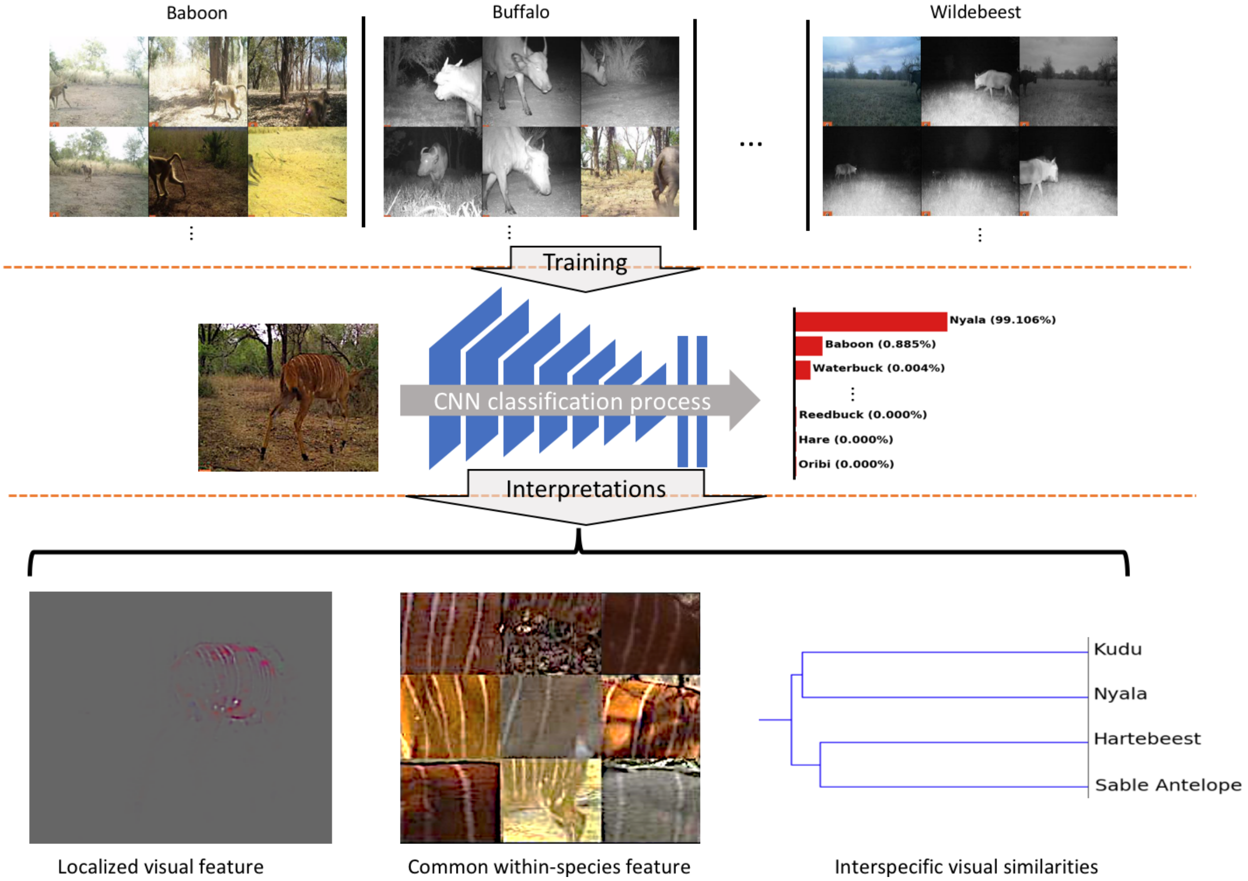


Figure 1.1: **Overview of training and interpretations.** We trained a standard CNN (VGG-16) on a camera trap dataset collected from Gorongosa National Park, Mozambique, to classify/recognize animals species. To interpret the mechanisms behind the recognition, first, we used Guided Grad-CAM (GG-CAM) to find the localized visual features extracted by the CNN from the images. Next, we used Mutual Information (MI) on the neurons of the last convolutional layer to generate common within-species features of each species. Finally, we used hierarchical clustering on the feature vectors to study the relative interspecific visual similarities of each species in the dataset.

of both known and unknown (i.e., novel) animals species to the CNN. The results imply that visual similarities can be used to identify visually distinct unknown animal species. In the Discussion section, we provide a brief example of how interpretations of CNNs can help understand the causes of misclassification and make potential improvements to the method. We also present results using a different CNN architecture (ResNet-50 [46]) to demonstrate the generalization of our observations.

# Background

## Camera trapping

Camera trapping has become an increasingly popular tool for remote wildlife monitoring [79]. It is a low-cost method for gathering data on an entire wildlife community and is especially useful for studying rare, cryptic, or nocturnal species, including many species of conservation concern [10]. Long-term camera trap datasets can be useful for monitoring trends in populations over time, and further analyses enable the estimations of relative densities and abundances across time and space [122]. Camera traps set across spatial gradients of environmental heterogeneity can also be used to understand environmental and anthropogenic drivers of wildlife distributions. Analyses can range from simple statistical tests (e.g., Analysis of Variance of relative activity across habitat types) to complex models (e.g., Bayesian hierarchical occupancy models that account for imperfect detection and incorporate multiple predictors of occupancy and detection [59, 103]). Finally, images or videos from camera traps provide insights into animal behavior, including movement and migration patterns, foraging and anti-predator vigilance, or reactions to experimental stimuli [13]. However, in all of these cases, camera trap studies are limited by the inefficiencies of manual data processing. Deep learning with CNNs has proven to improve the efficiency of relevant studies drastically. In this chapter, we interpret the mechanisms of deep learning in detail.

## Deep learning in camera trap classification

Deep learning is a subdomain of machine learning that uses algorithms inspired by biological neural networks [65]. It has gained much attention among ecologists in recent years [144], with animal species identification from camera trap images using CNNs being one of the most popular applications [18, 85, 137, 114, 123, 140]. Chen et al. [18] made the first attempt to classify camera trap images with deep learning methods automatically. They achieved only 38% test accuracy on their 20,000-image dataset and suggested that, with enough training data, deep learning can surpass other methods. Gomez et al. [137] harnessed deep learning with transfer learning, a method of model fine-tuning, to identify animal species in the Snapshot Serengeti dataset [148, 124] and achieved over 80% test accuracy using large amounts of data. Further, Norouzzadeh et al. [85] trained multiple CNN architectures on a similar dataset as Gomez et al. and achieved a test accuracy of over 90%. However, to our knowledge, there are no studies explicitly explaining the mechanisms of deep learning that facilitate the classification of animals.

## Basic mechanisms of CNNs

Convolutional neural networks (CNNs) are one of the most frequently used deep networks in computer vision and deep learning. From AlexNet [60] to VGG [115] and ResNet [44], the capacity of modern CNN architectures has advanced rapidly, resulting in high recognition performance that makes various real-world applications possible. Modern CNN architectures typically have three types of layers—-convolutional layers, pooling layers, and fully-connected layers-—which gradually transform an input image into a predicted category label. For instance, the VGG-16 network (the architecture used in this paper) has 13 convolutional layers, 5 pooling layers, and 3 fully-connected layers. Convolutional layers in CNNs consist of local filters or neurons and are designed to capture spatially-distributed local traits such as edges, parts, and textures [150]. Pooling layers account for the larger receptive fields of the deeper convolutional layers, i.e., the subsequent convolutional layers assemble the previously learned local traits into more globally-perceived shapes and configurations [153]. Finally, fully-connected layers abstract all of the local and global traits/features into high-level semantic concepts like categories and attributes [73]. All the parameters in the CNNs are learned by minimizing the errors between predictions and ground-truthed labels through a layer-by-layer updating process called back-propagation. In this chapter, we focus on interpreting the inner representations of CNNs and examining the relationship between neurons and ecological data.

### Interpretable deep learning

Though deep learning has achieved impressive performance on many visual recognition tasks, its "black-box" mechanism makes it hard for users to understand the underlying recognition process. To alleviate these drawbacks, researchers have developed various methods towards interpretable deep learning by decomposing and organizing the internally learned features. Representative works include GG-CAM [111] and network dissection [152], which localize and extract meaningful parts and regions that are coherent to human perception and reasoning [63]. The interpretable deep learning techniques have been successfully applied to face analysis [73], scene understanding [153], chest radiograph diagnosis [98], and plant species identification [110, 141]. In this chapter, we leverage the recent advances in interpretable deep learning to shed light on deep learning-based wildlife recognition and provide useful practices and new insights for future deployment in ecological fields.

## Data

The dataset used in this chapter was collected from Gorongosa National Park, Mozambique. It contains a total of 30 animal species. We used data from the 20 most commonly photographed mammal species to train our model for higher training performance and more accurate feature extraction. The 20 species include:

1. `Buffalo` (African buffalo, *Syncerus caffer*)

2. `Elephant` (African elephant, *Loxodonta africana*)

3. `Hare` (African savanna hare, *Lepus microtis*)

4. `Baboon` (*Papio cynocephalus*)

5. `Wildebeest` (Blue wildebeest, *Connochaetes taurinus*)

6. `Bushpig` (*Potamochoerus larvatus*)

7. `Bushbuck` (Cape bushbuck, *Tragelaphus sylvaticus*)

8. `Civet` (*Civettictis civetta*)

9. `Reedbuck` (Southern reedbuck, *Redunca arundinum*)

10. `Porcupine` (Crested porcupine, *Hystrix cristata*)

11. `Kudu` (Greater kudu, *Tragelaphus strepciseros*)

12. `Impala` (*Aepyceros melampus*)

13. `Genet` (Large-spotted genet, *Genetta tigrina*)

14. `Hartebeest` (Lichtenstein's hartebeest, *Alcelaphus buselaphus*)

15. `Nyala` (*Tragelaphus angasii*)

16. `Oribi` (*Ourebia ourebi*)

17. `Sable Antelope` (*Hippotragus niger*)

18. `Vervet Monkey` (*Chlorocebus pygerythrus*)

19. `Warthog` (*Phacochoerus africanus*)

20. `Waterbuck` (*Kobus ellipsiprymnus*).

The total number of images of the 20 most common animal species is 111,467, and the distribution of the 20 species is illustrated in Figure 1.2. There are 6,596 images (around 6% of the dataset) with multiple animal species in the same scene (i.e., multi-labeled). We do not have specific preprocessing for the multi-label images because we want the whole training process to be as realistic as possible, where misclassifications caused by current data storage protocols of multi-label data can happen. Details of data background and training-validation-testing split are reported in Appendix A.

The rest of 10 species were used solely for performance evaluation purposes. These 10 species are:

1. `Aardvark` (*Orycteropus afer*)

2. `Bushbaby` (Brown greater galago, *Otolemur crassicaudatus*)

3. `Eland` (*Taurotragus oryx*)

4. `Honey Badger` (*Mellivora capensis*)

5. `Lion` (*Panthera leo*)

6. `Samango` (*Cercopithecus albogularis*)

7. `Serval` (*Leptailurus serval*)

8. `Ground Hornbill` (Southern ground hornbill, *Bucorvus leadbeateri*)

9. `Pangolin` (Temminck's ground pangolin, *Smutsia temminckii*)

10. `Rodent` (multiple rodent species).

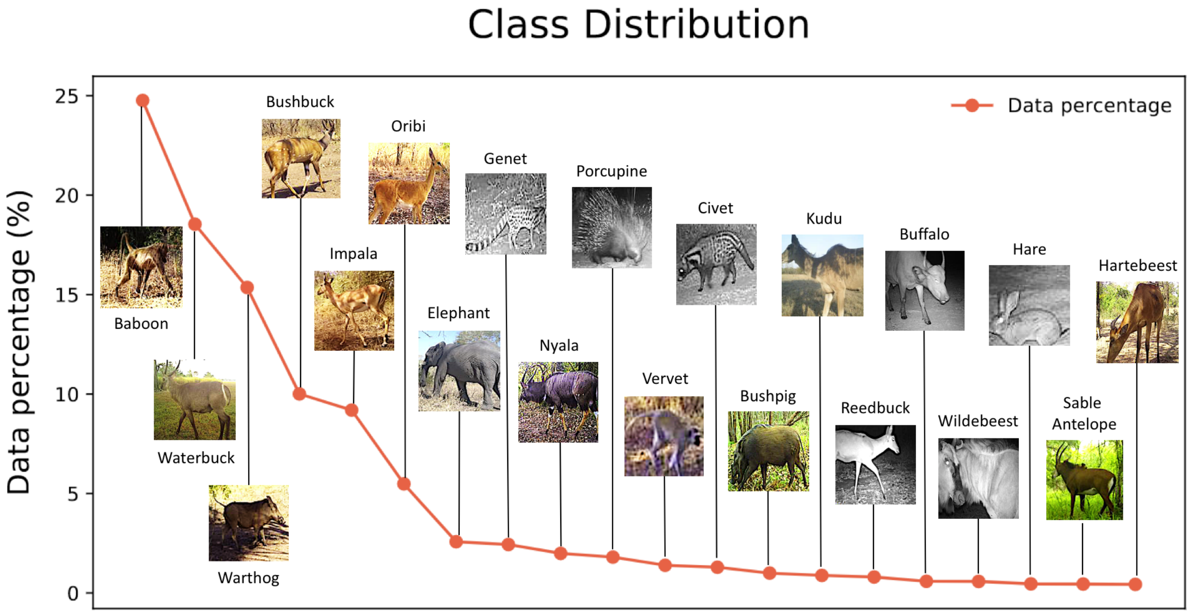


Figure 1.2: **Distribution of the 20 most abundant animal species.** More than 60% of the images are from the first three species.

# Model interpretations

Before interpreting a CNN, we firstly trained a standard CNN (VGG-16 [115]) on the 20 most abundant species. Figure 1.3 shows the test performance of our trained model. The average per-class test accuracy was 83.0%, ranging from 95.2% for `Civet` to 54.3% for `Reedbuck`.

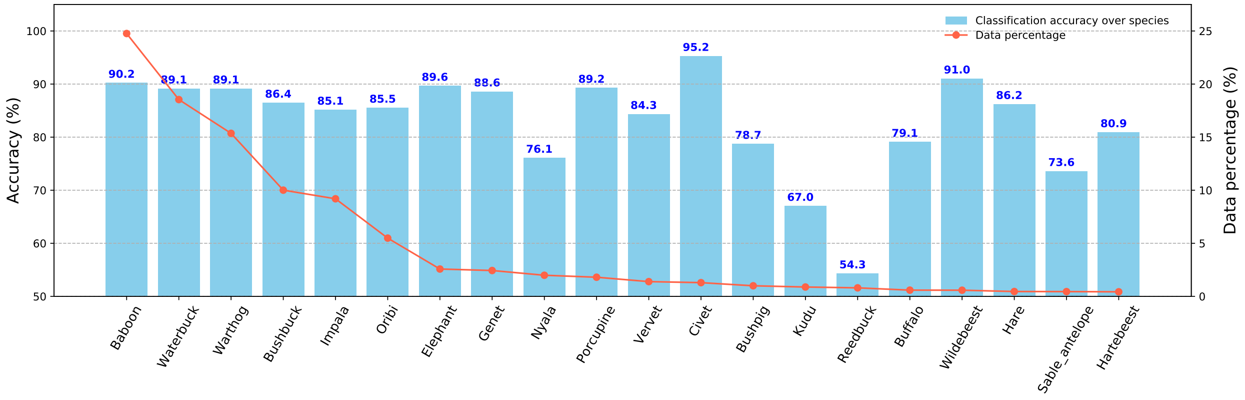


Figure 1.3: **Animal classification accuracy and data distribution per-species.** `Reedbuck` and `Kudu` are the two species that had the lowest test accuracy (54.3% and 67.0%, respectively).

## Interpretation 1: Localized feature visualization

After the model was trained, we used GG-CAM, a method that combines the outputs of GBP [119] and Grad-CAM [111], on the last convolutional layer of our model (see Appendix A for implementation details). Specifically, in our experiment, Grad-CAM captured the most discriminative image patches, GPB captured visual features both within and outside of the focal Grad-CAM patches, and, in combination, GG-CAM captures the features most salient to the actual recognition process (Figure 1.4). When making correct classification, the CNN could extract species-specific features from the input images, such as the white spots and the white stripes of the `Nyala` image in Figure 1.4.

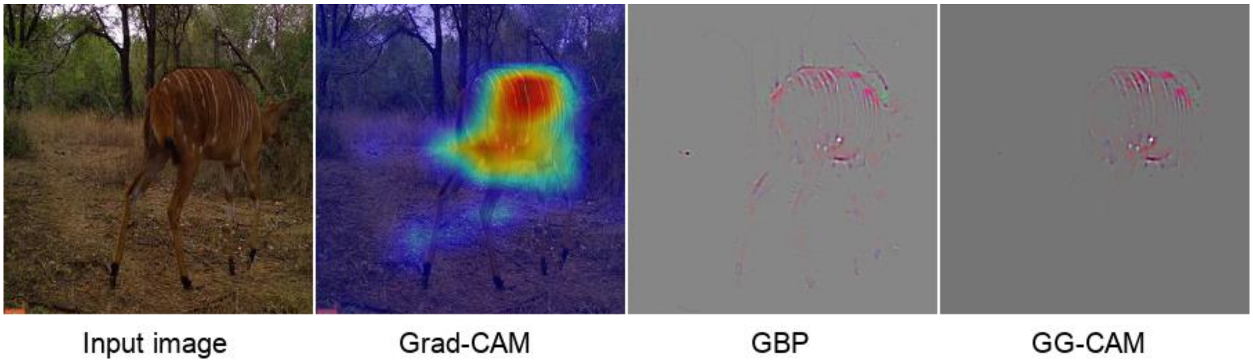
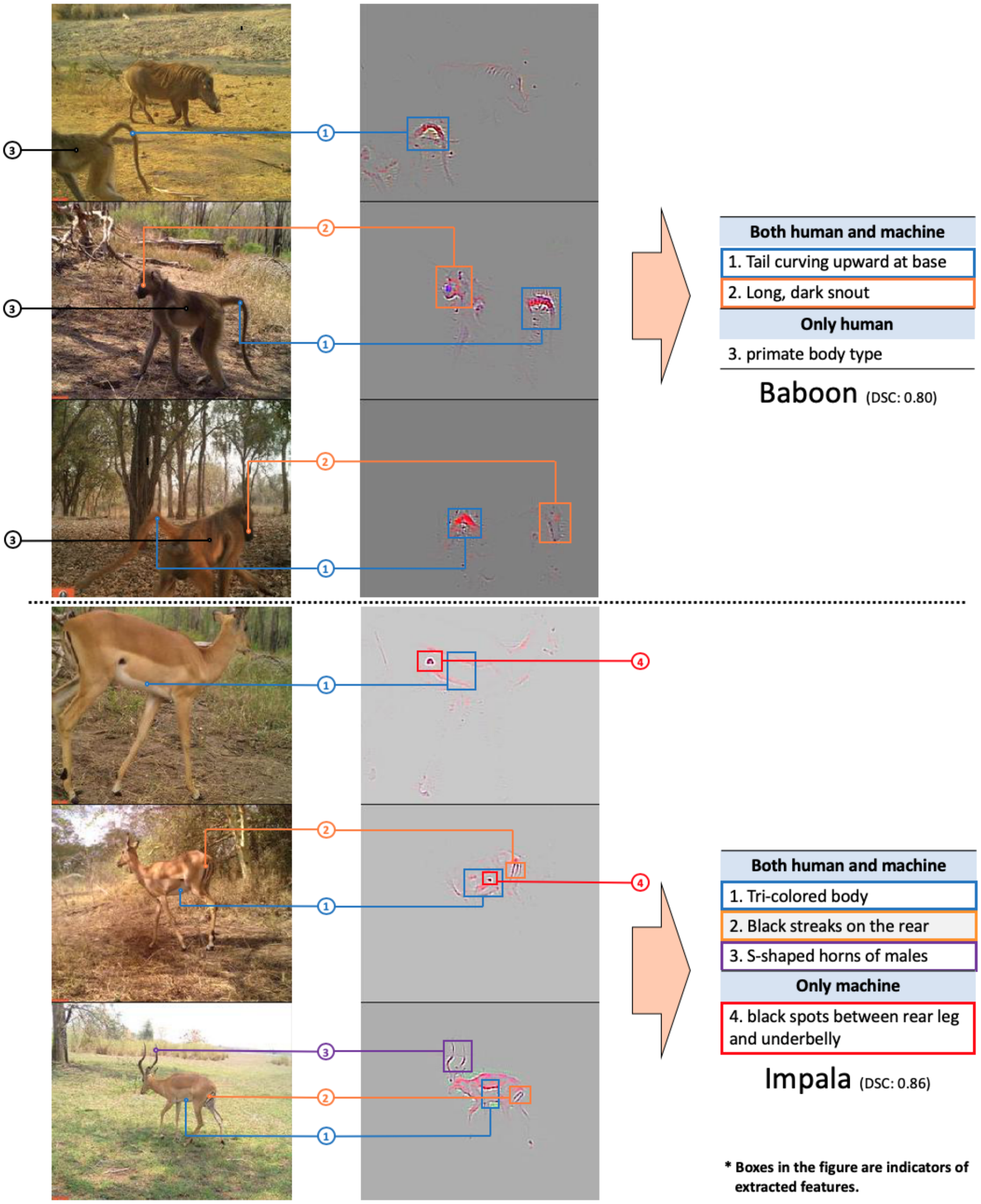


Figure 1.4: **Comparison of outputs from Grad-CAM, GBP, and GG-CAM.** Once trained, any image (leftmost panel) can be overlaid with its Grad-CAM heat map (left center panel) to identify the region of interest to the CNN. GBP identifies the most important visual features in the pixel space to our CNN (center right panel), and its output can be weighted by the Grad-CAM heat map to produce GG-CAM output seen in the rightmost panel. Note that in this `Nyala` image, GBP is less discriminative than GG-CAM: both highlight the stripes of the nyala, whereas GPB includes non-species-discriminative tree branches and legs.

Next, we inspected the GG-CAM outputs in order to obtain information on the reasoning of deep learning classification [153, 151]. We conducted a relatively informal assessment of similarity matching of the features extracted by GG-CAM to those most salient to human experts (i.e., visual descriptors, Figure 1.5). The matching was agreed upon by at least 2 of 4 authors (ZM, KMG, ZL, and MSN). Details of the generation of the human visual descriptors are reported in Appendix A. Figure 1.5 shows that to some extent, the trained CNN used features similar to those used by experts to identify animal species in our images. Specifically, Figure 1.5-Baboon shows that our CNN used faces and tails to identify `Baboon` images. Both of the two features were also used by human experts to identify `Baboon`. In Figure 1.5-Impala, besides the black streaks on the back ends, the line separating the colors of the upper body from the white underbelly, and the S-shaped horns, the CNN also considered the black spots between the rear legs and bellies of `Impala` as a discriminative feature. Although not included as one of the most-used human visual descriptors, this feature is a good example of a feature traditionally overlooked by humans but now identified by our CNN as salient. More challenging examples of `Reedbuck` can be found in Figure 1.6. They indicate that the poor test performance of `Reedbuck` (54.3%) can be caused by the lack of discriminative visual features. We also calculated the Dice Similarity Coefficient (DSC) [118] between machine extracted features and human visual descriptors to provide a quantitative sense of how similar these two sets of features are for each species. The closer the DSC value is to 1, the more similar machine extracted features are to human visual descriptors. (Figure 1.5 and 1.6; Appendix A).



1
2
3
4
Both human and machine
1. Tail curving upward at base
2. Long, dark snout
Only human
3. primate body type
Baboon (DSC: 0.80)
Both human and machine
1. Tri-colored body
2. Black streaks on the rear
3. S-shaped horns of males
Only machine
4. black spots between rear leg and underbelly
Impala (DSC: 0.86)
* Boxes in the figure are indicators of extracted features.

Figure 1.5: **GG-CAM generated localized discriminative visual features of randomly selected images of `Baboon` and `Impala`.** For `Baboon`, the CNN focused on faces and tails. For `Impala`, the CNN used the contrast between the white underbelly and dark back, black streaks on the rear, and black spots between the rear legs and underbelly. Most of the features extracted by the CNN had counterparts (similar focal visual components) in the human visual descriptors (indicated by the colors and agreed upon by at least 2 of 4 authors).

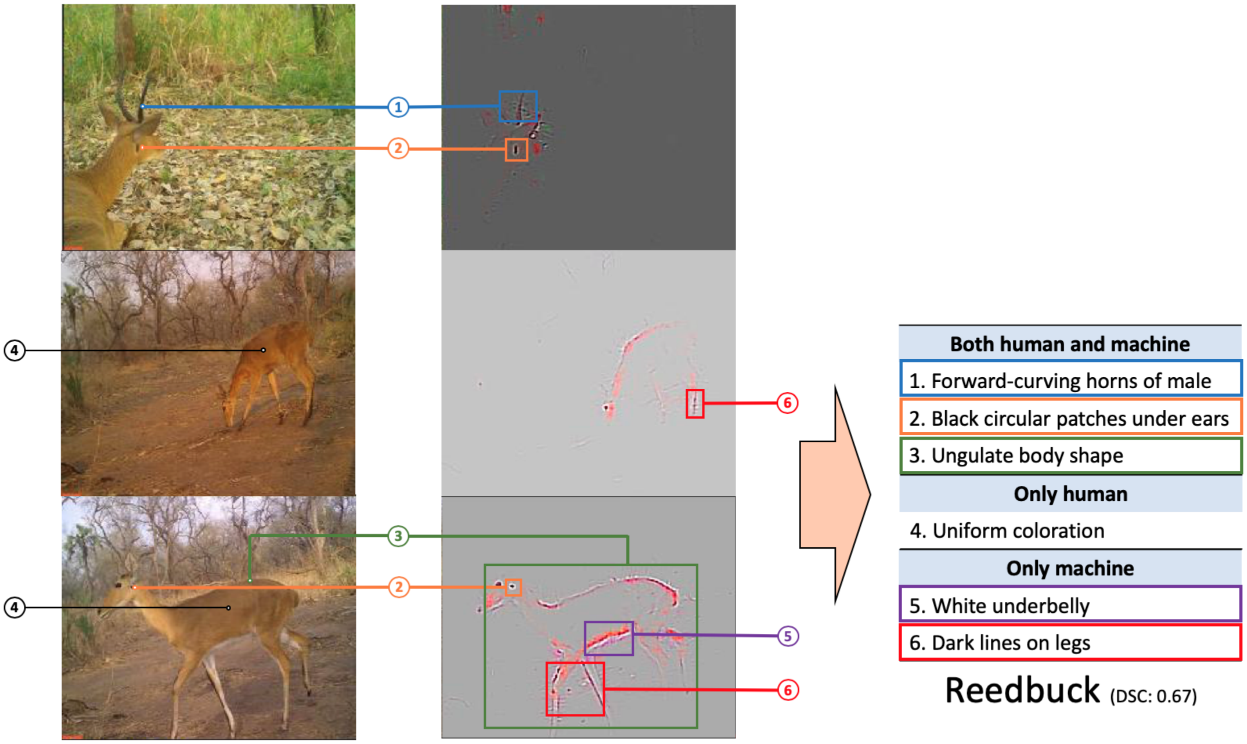


Figure 1.6: **Localized features of `Reedbuck`.** The extracted features of `Reedbuck` images (e.g., ungulate body shape, black circular patches under ears, white underbelly, and dark lines on legs) were not as discriminative compared with features of other species, such as the black stripes of `Impala` and white stripes of `Nyala`. This might also be the reason why the class accuracy of `Reedbuck` was relatively poor.

## Interpretation 2: Common within-species features

Next, we used Mutual Information (MI) [8, 78] to extend the features of single images to within-species features of each animal species. We calculated the MI scores for each of the neurons in the last convolutional layer of our CNN to indicate their importance to all images of one of the selected species (implementation details are reported in Appendix A). In short, for each of these neurons, we obtained 20 species-specific MI scores from 6,000 randomly selected training images (300 images of each species). For each species, we identified the 5 neurons in the last convolutional layer that produced the 5 highest scores. We then identified the top 9 "hottest" 60x60 pixel patches (within-species features) to which each of these top 5 neurons responded in each image (e.g., Figure 1.7). These features generalized across all images within the same species (see Appendix A for examples of common within-species features of all species). Most results were associated with distinguishable visual features of the animals, for example, black spots of `Civet`, trunks of `Elephant`, quills of `Porcupine`, and white stripes of `Nyala`.

However, visual similarities of animal species are not the only information our CNN used to identify species. The CNN also used information such as trees in the background to identify species frequenting woodlands, especially when most of the images were from similar environments or the same camera trap locals (e.g., image patches of the top 1 neuron of `Porcupine` in Figure 1.7). `Reedbuck` in Figure 1.7 is another good example. Image patches of the top 4 neuron are mostly the same. This is because many of the `Reedbuck` images were taken by the same camera, which produced common backgrounds. This information reflects the inner bias of the dataset. For example, when most of the images of a class were taken from similar camera locals (i.e., backgrounds of the images could be similar), our model did not have to learn species-specific features during training, and the generality of the CNN was largely degraded [131]. Enhancing CNN's ability to focus more on target objects/animals is a future direction to improve the generality of animal classification.

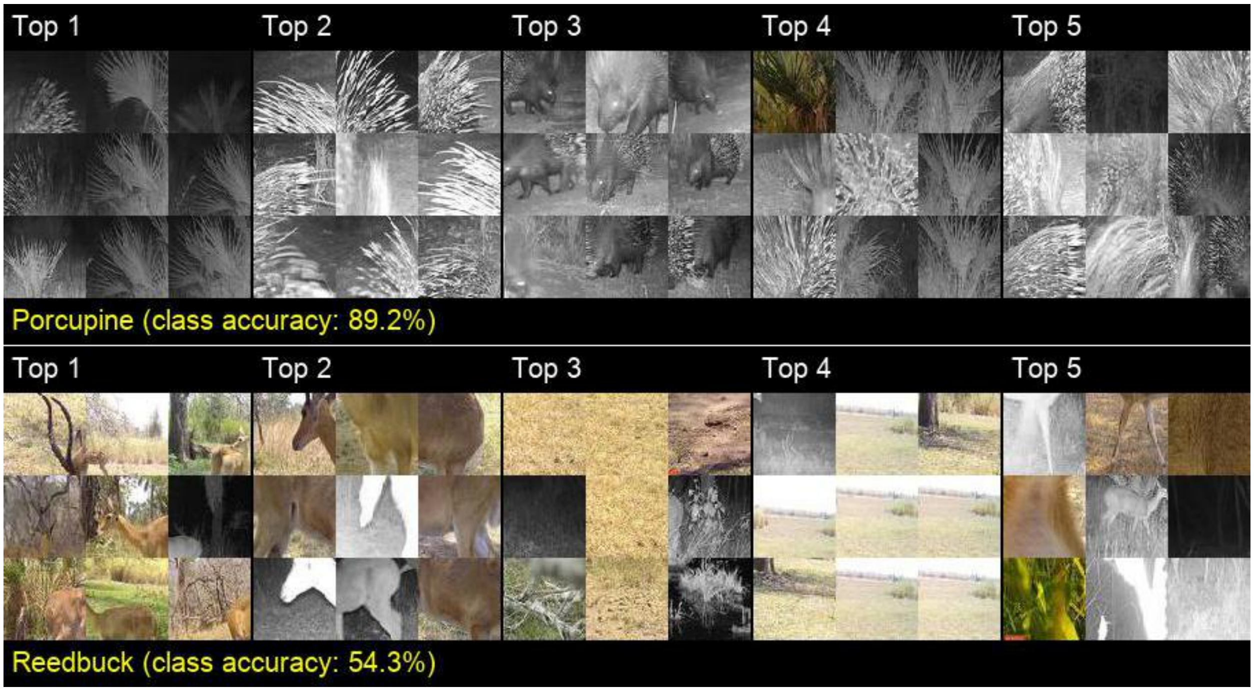

Top 1
Top 2
Top 3
Top 4
Top 5
Porcupine (class accuracy: 89.2%)
Top 1
Top 2
Top 3
Top 4
Top 5
Reedbuck (class accuracy: 54.3%)

Figure 1.7: **Image patches that responded most strongly to the 5 neurons with the highest MI scores of `Porcupine` and `Reedbuck`.** For each row/species, the leftmost set of 9 60x60-pixel patches was extracted from 9 camera trap images that include a species of interest and had the highest responses to the corresponding neuron. In each of the 9 cases, the extracted patches were centered around the "hottest" pixel (i.e., highest response) of the neuron (in the last convolutional layer of our CNN) that had the highest MI score for the said species class. The MI scores were calculated using 6,000 randomly selected training images (300 images per class). The remaining 4 sets of 9 patches were equivalently extracted for the neurons with the next 4 highest MI scores. These patches provide a sense of the within-species features to which the neuron in question responded. The higher the class accuracy, the more closely correlated these image patches are for the species of interest. For example, in the relatively accurately classified `Porcupine` images (89.2% test accuracy), the first neuron (Top 1 of the upper row) responded to palm plants that appeared in most of the training images that also contained porcupines. The second neuron (Top 2) responded to the quills, while the third neuron (Top 3) responded most strongly to bodies with faces. On the other hand, in a much less accurately identified `Reedbuck` set, the first neuron (Top 1 of the lower set) responded to branch-like structures, including tree limbs and horns. The patterns in these patches are less consistent than the `Porcupine` patches. Note that some sets of patches are primarily backgrounds (e.g., Top 1 upper row and Top 4 lower row), from which we can infer that our CNN learned to associate certain backgrounds with particular species. Such associations, however, only arise because particular cameras produce common backgrounds for all their images, thereby setting up a potential for a camera-background and species correlation that can well disappear if additional cameras are used to capture images. Similar sets of images are illustrated for all other species in Appendix A.

## Interpretation 3: Interspecific visual similarities

Then, we generated a visual similarity dendrogram for all species by applying hierarchical clustering [105] to the CNN feature vectors of 6,000 randomly selected training images, i.e., the outputs of the last fully-connected layer (which is of dimension 4,096 in Euclidean space) of our trained CNN (see Appendix A for implementation details). This dendrogram (Figure 1.8) is an abstract representation of how images of species are separated in the feature space of our CNN. It also provides a means for quantifying how visually similar the 20 animal species are to our trained CNN. In the dendrogram, similar animals are measurably closer together than visually distinct ones (e.g., striped versus spotted; long-tailed versus no-tail), irrespective of their phylogenetic distance. Thus, though most of the antelopes are grouped (from `Sable Antelope` to `Reedbuck`), the large bull-like herbivores (`Wildebeest` and `Buffalo`) and pig-like mammals (`Warthog`, `Porcupine`, and `Bushpig`) are also grouped together even though they may belong to different families or orders (Figure 1.8).

A well-learned feature vector space can also help identify images that differ in some way from those on which the CNN has been trained (i.e., known v.s. unknown) [62, 138]. To examine the difference between known and unknown species to our CNN, we incorporated the 10 excluded rarer animal species (i.e., unknown species) into the testing data. Then we implemented a 10-round random selection of each species to measure the relative unfamiliarity of both known and unknown species to our CNN. Specifically, in each round, we randomly selected 20 testing images of the 30 animal species and then calculated the Euclidean distances of their feature vectors to the 20 feature centroids of known species (calculated using training samples). The relative unfamiliarity of each class was calculated as the mean distance of the 20 testing images to their closest feature centroids across the 10-round random selection (Fig. 1.9). The intuition is that the more familiar the species are to the network, the closer the average distances are to 1 of the 20 feature centroids of training data. The known species had relative unfamiliarity values ranging from 0.95 to just over 1.1, with `Elephant` being the largest at 1.14. We set this `Elephant` value to be our nominal unfamiliarity threshold and found that 7 of the 10 species fell above it (i.e., were less familiar to our trained CNN than any of the known species; viz., `Pangolin`, `Honey Badger`, `Serval`, `Bushbaby`, `Rodent`, `Ground Hornbill`, and `Lion`). Three of the unknown species (viz., `Samango Monkey`, `Aardvark`, and `Eland`) that were considered familiar to our model appear to share features with the 20 known species (e.g., monkeyness: `Samango Monkey` unknown and `Vervet Monkey` known; antelopeness: `Eland` unknown, `Hartebeest`, `Wildebeest`, and `Sable Antelope` known).

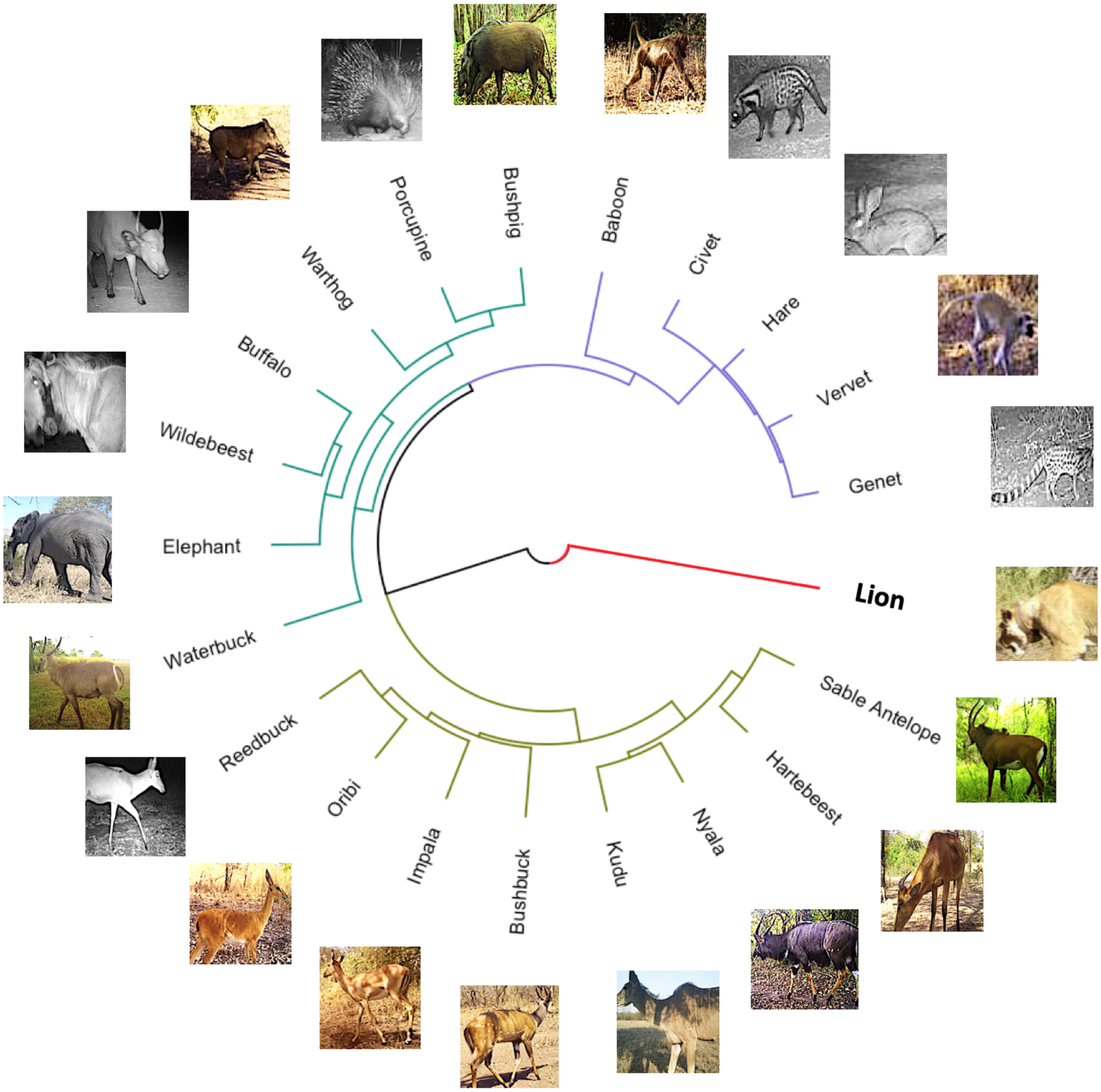
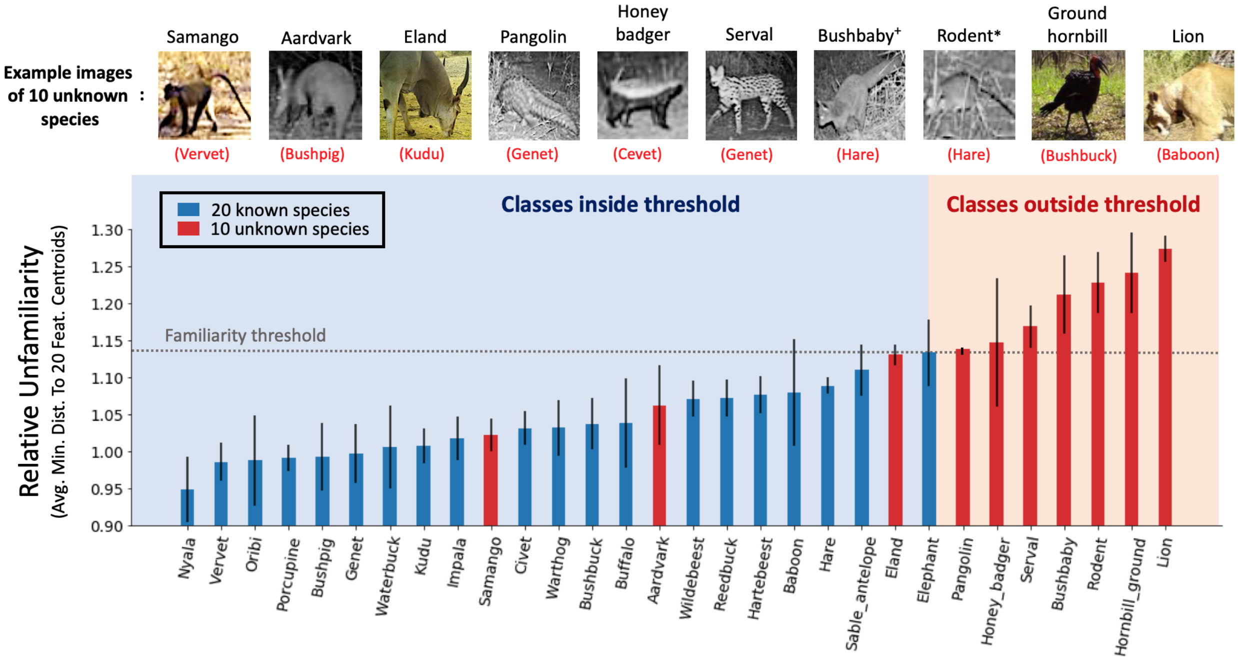


Figure 1.8: **Visual similarity dendrogram/tree of the feature space of our trained CNN.** The similarity tree is based on hierarchical clustering of the feature vectors of the last fully-connected layer of our CNN. The leaves represent feature vector centroids of 300 training images of each species, and their relative positions in the tree indicate the Euclidean distances between these centroids in the feature space. In the similarity tree, the more similar two species are to our CNN, the more tightly coupled they are in the tree. For example, green, purple, and brown branches correspond to three primary clusters that appear to be a small to medium-sized antelope cluster, an animal-with-prominent-tail or big-ears cluster (though `Baboon` seems to be an outlier in this group), and a relatively large body-to-appendages group (with `Waterbuck` as the outlier in this group). When the feature vectors of unknown animal species are placed in the tree (e.g., the red branch of `Lion`), sometimes they can differ significantly from those of the known species.



+ Bushbaby represents brown greater galago.
* Rodent represents small rodent animals that could not be identified as a single species.
· Red labels in parentheses are species that most of the images of the unknown species are classified as.

Figure 1.9: **Relative unfamiliarity of the 30 species (including the 10 unknown species) to the CNN.** 20 species were used to train the CNN (known species—see Fig. 1.8), and then 10 additional species (unknown species) were tested to see how their average feature vectors (averaged across 20 different exemplar photographs for each species) fell within the feature vector space. 7 of the 10 unknown species had average feature vectors yielding a relative unfamiliarity value above our nominal unfamiliarity threshold. The threshold is defined as the known species having the highest relative unfamiliarity value (`Elephant` in our experiment).

# Discussion

## Misclassifications

Understanding the mechanisms of deep learning classifications of camera trap images can help ecologists determine the possible reasons for misclassifications and develop intuitions of deep learning, which is necessary for method refinement and further development. For example, Figure 1.3 shows that `Reedbuck` was the least accurately classified species by the CNN. The classification confusion matrix [32] (see Appendix A) reveals that many `Reedbuck` images were classified as `Oribi` (8%), `Impala` (12%), and `Bushbuck` (12%). On the other hand, Figure 1.8 shows that `Reedbuck` is close to `Oribi`, `Impala`, and `Bushbuck` in the dendrogram, which partly explains the reasons for the misclassifications because the CNN considered these 4 species similar to each other. Further, the localized visual features of the misclassified images can provide more detailed information on possible reasons for misclassifications. For example, Figure 1.10 shows that although the CNN could locate the animals in most of the images, it was challenging for the CNN to classify the images correctly when the distinctive features of the species were obscured or multiple species were in the same scenes.

## Comparisons with ResNet-50

Finally, we present the results using ResNet-50 [45], another widely used CNN architecture in computer vision with much more layers but fewer parameters than VGG-16, to demonstrate the generalization of our observations. The classification performance is reported in Table 1.1, where ResNet-50 was slightly better than VGG-16, with a 0.5% improvement on the micro accuracy and a 0.2% improvement on the macro accuracy.

Table 1.1: **Test accuracy of ResNet-50 and VGG-16**

| Metric | Accuracy (ResNet-50) | Accuracy (VGG-16) |
|---|---|---|
| Overall micro accuracy: | 88.1 % | 87.5 % |
| Overall macro accuracy: | 83.2 % | 83.0 % |

Next, we performed the GG-CAM analysis using the trained ResNet-50 model on the same input images in Figure 1.5. From Figure 1.11, we observe that both ResNet-50 and VGG-16 extracted similar visual features. ResNet-50, as mentioned in the original paper [45], was more sensitive to edges compared to VGG-16, and for the `Baboon` images, it was less sensitive to the warm colors.

Then, we used the same set of images for the VGG-16 MI experiments to generate the MI results for ResNet-50. By comparing the image patches (Figure 1.12), we observe that, although different by appearance, ResNet-50 also identified quills (Top 3 and Top 5) and heads (Top 1) of `Porcupine`. For `Reedbuck` images, the extracted features were relatively random, and the class accuracy was 1.2% lower than VGG-16.

Finally, we used the same set of images used for interspecific similarity experiments to generate hierarchical clustering results for ResNet-50. ResNet-50, however, appears to be

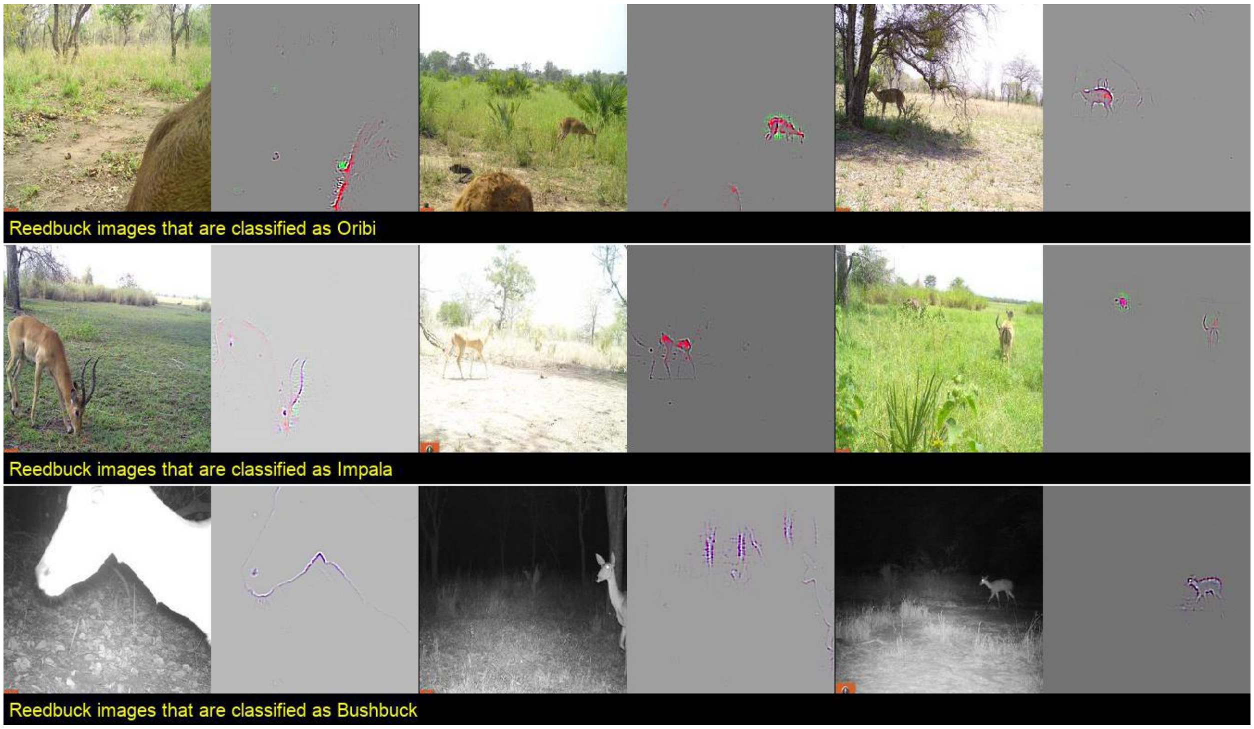

Reedbuck images that are classified as Oribi
Reedbuck images that are classified as Impala
Reedbuck images that are classified as Bushbuck

Figure 1.10: **Examples of `Reedbuck` images that were misclassified as `Oribi`, `Impala`, and `Bushbuck`, with corresponding localized visual features.** These examples show that although the CNN could locate the animals in most images, it was hard for the machine to find distinctive features from: 1) images with animals that were far away in the scene; 2) over-exposed images; 3) images that captured only parts of the animal; and 4) images with multiple animal species. In many of these images, the misclassified species are indeed present in the scenes and are often in the foreground. This problem is an artifact of the current labeling process and remains to be resolved in the future. For example, the animal in the leftmost image on the second row that was classified as `Impala` is an `Impala`. The CNN correctly classified this image based on the animal. However, this image was labeled as `Reedbuck` because the extremely small black spots far in the background are `Reedbuck`. When two species appear in the same scene, the same image was saved twice in the dataset with different labels corresponding to different species in the scene. This labeling protocol is common in camera trap programs but can confuse CNNs and remains a problem that must be resolved in the future.

not as good as VGG-16 in forming an antelope group (red branches) of species since both `Baboon` and `Warthog` are in this antelope group.

These three comparative scenarios show that ResNet-50 and VGG-16 did not have markedly different performances in learning the visual features for wildlife classification.

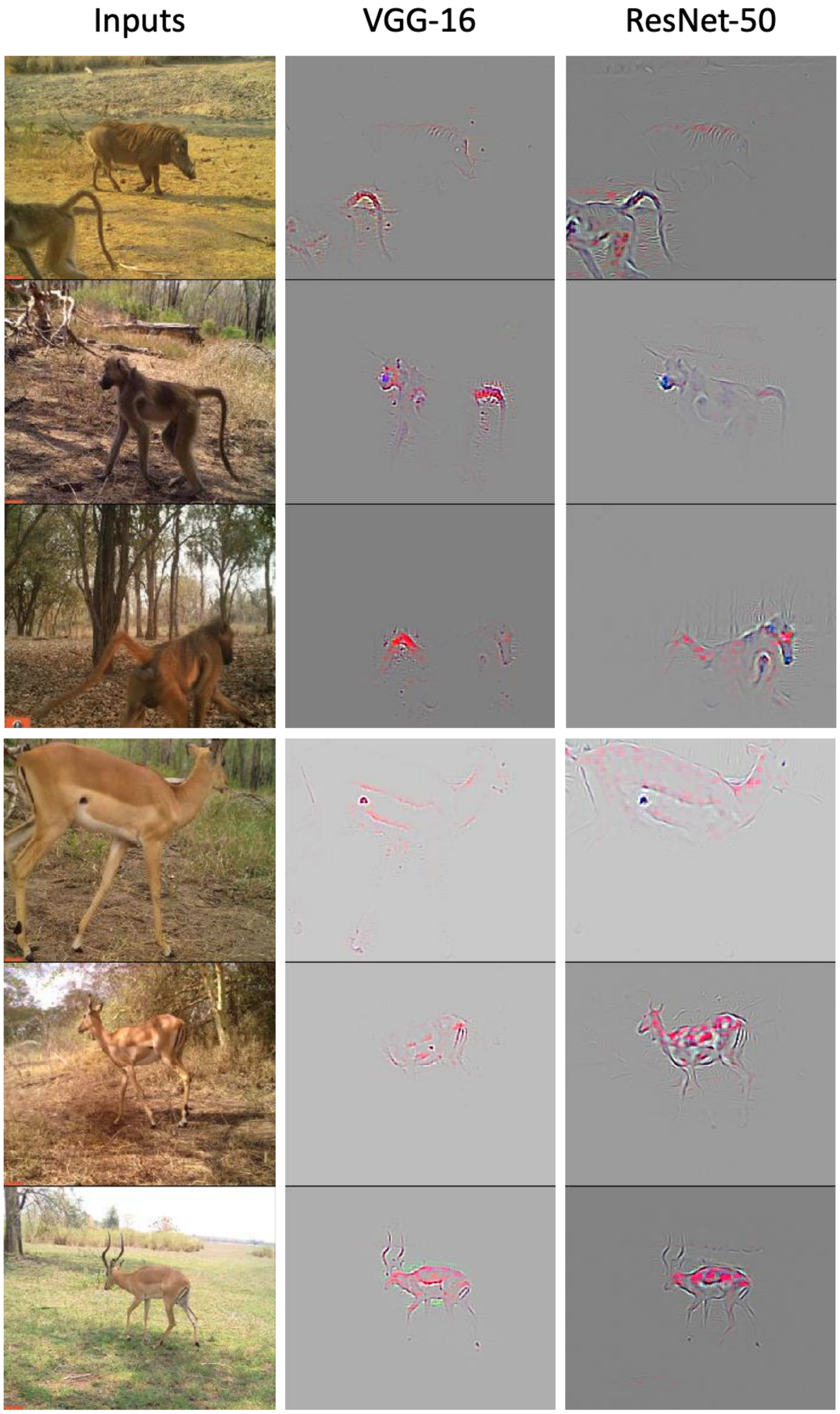


Figure 1.11: **Comparison of localized features of VGG-16 and ResNet-50.** Although both models extracted similar features from the same images, ResNet-50 was more sensitive to the edges of target objects.

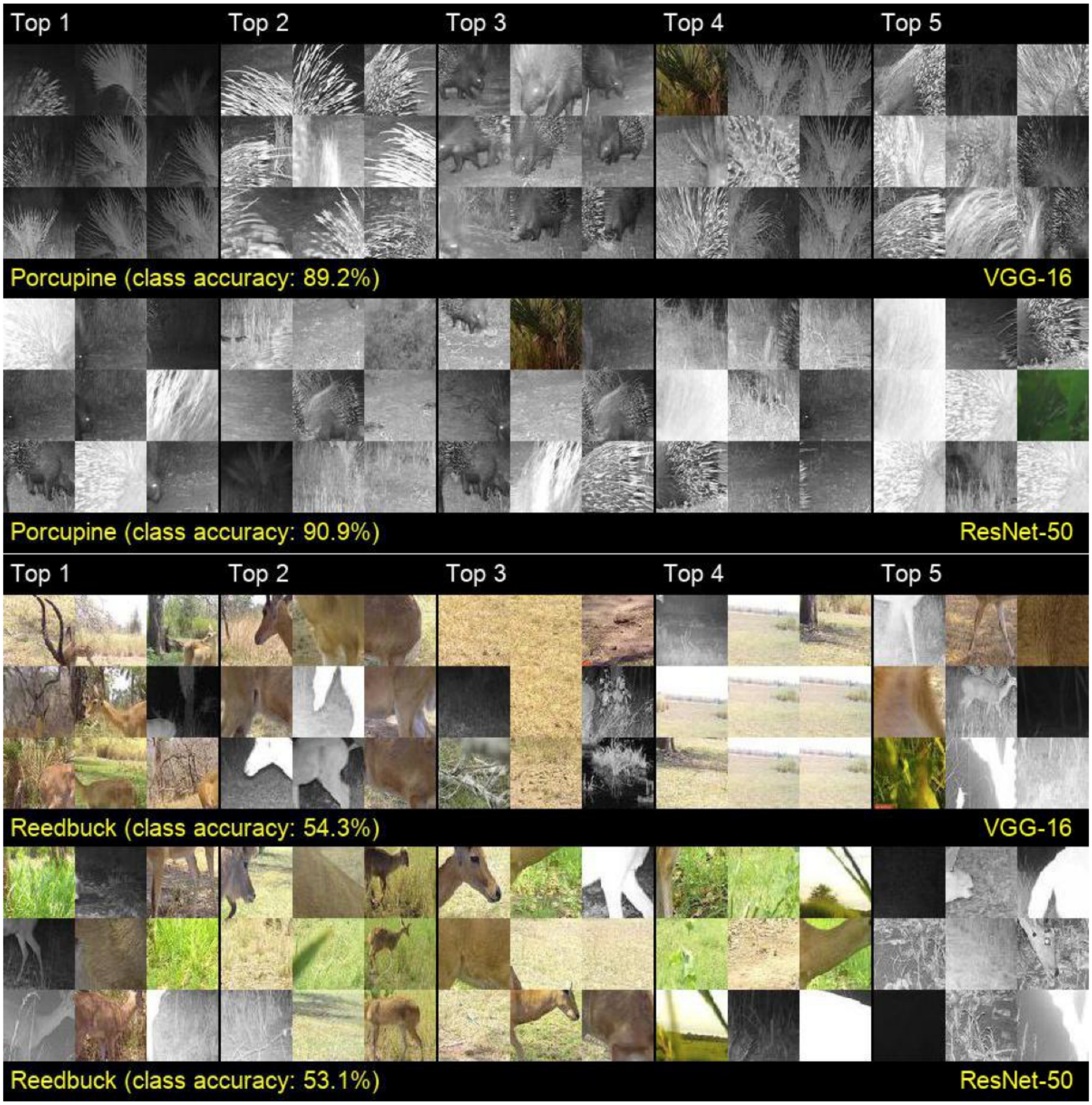

Top 1
Top 2
Top 3
Top 4
Top 5
Porcupine (class accuracy: 89.2%)
VGG-16
Porcupine (class accuracy: 90.9%)
ResNet-50
Top 1
Top 2
Top 3
Top 4
Top 5
Reedbuck (class accuracy: 54.3%)
VGG-16
Reedbuck (class accuracy: 53.1%)
ResNet-50

Figure 1.12: **Comparison of the MI results of VGG-16 and ResNet-50.** ResNet-50 was also able to extract quills and heads of `Porcupine`. For `Reedbuck` images, ResNet-50 focused on relatively random elements in the images.

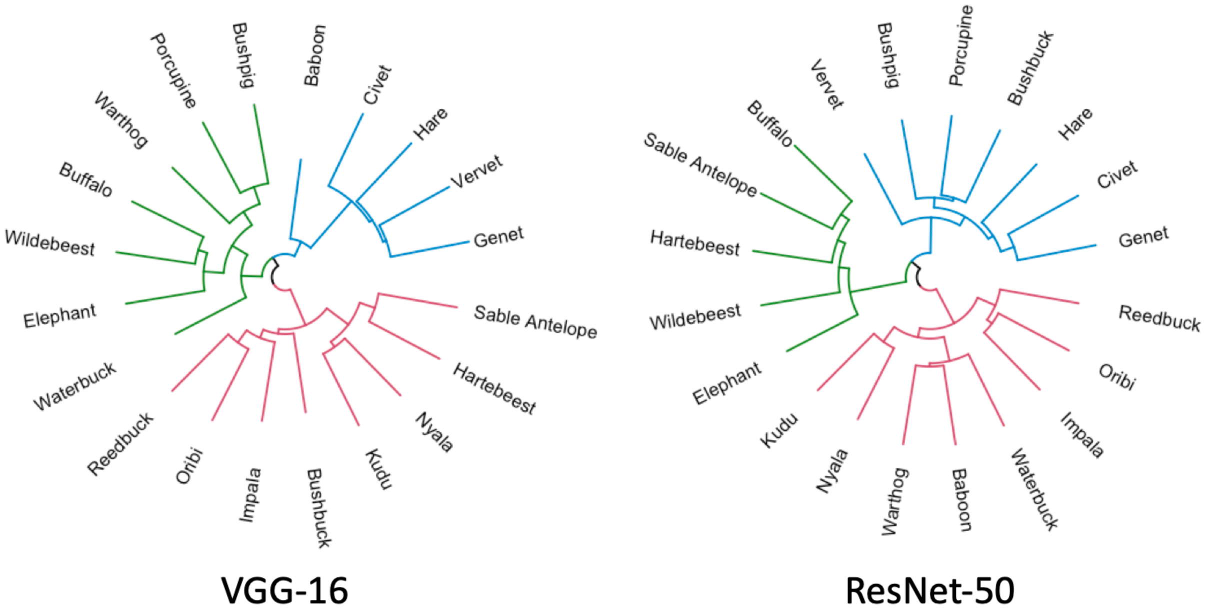


Figure 1.13: **Comparison of hierarchical clustering results of VGG-16 and ResNet-50.** The composition of the classes is somewhat different between the VGG-16 and ResNet-50 dendrograms, with the former having a more coherent antelope group (red branches) than the latter.

# Conclusion

Deep learning has become a core component of data science and fields using big data. Ecology has been no exception [55, 99]. This shift requires that new methods, including models from machine learning and artificial intelligence, are accessible and usable by ecologists [22]. This chapter provides practical steps in model interpretation to help ecologists take advantage of deep learning as a cutting-edge approach for future research and overcoming major methodological roadblocks. The interpretations described in this chapter are steps toward a more informed use of deep learning methods. For example, future research involving the training of CNNs to identify individuals in ecological studies, whether for purposes of species classification, conservation biology, sustainability management, or identification of specific individuals in their own right [51, 61] (e.g., in behavioral studies) can follow the methods presented here to identify the sets of features being used to classify individuals. This information may then be used in creative ways yet to be imagined to improve CNN training and, hence, raise the level of performance of CNNs as an aid to analyzing ecological data.

# Chapter 2

# Challenges and approaches of avian recognition from aerial images

Zhongqi Miao[1,2], Stella X. Yu[1,2], Kyle L. Landolt[3], Mark D. Koneff[4], Timothy P. White[5], Luke J. Fara[3], Enrika J. Hlavacek[3], Bradley A. Pickens[6], Travis J. Harrison[3], Wayne M. Getz[1,7]

This chapter is in preparation for journal submission and is included as a dissertation chapter with permission from co-authors.

[1]Dept. Env. Sci., Pol. & Manag., UC Berkeley, CA, United States
[2]International Comp. Sci. Inst., UC Berkeley, CA, United States
[3] Upper Midwest Env. Sci. Cent., USGS, La Crosse, Wisconsin, MN, United States
[4] Div. of Migratory Bird Manag., US Fish & Wildlife Service, Orono, ME, United States
[5] Env. Studies Prog., Bureau of Ocean Energy Management, Sterling, VA, United States
[6] Div. of Migratory Bird Manag., US Fish & Wildlife Service, Laurel, MD, United States
[7]Sch. Math. Stat. Comp. Sci., Univ. KwaZulu-Natal, South Africa

## Abstract

Remote aerial sensing provides a non-invasive, large-geographical-scale technology for avian monitoring, but the manual processing of images limits its development. Artificial intelligence methods can be used to mitigate this manual image processing requirement. The implementation of AI methods, however, is limited by several challenges: 1) imbalanced (i.e., long-tailed) data distribution, 2) annotation uncertainty in categorization, and 3) dataset discrepancies across different study sites. Here we use aerial imagery data of waterbirds around Cape Cod and Lake Michigan to examine how these challenges limit avian recognition performance. We review existing solutions and demonstrate as use cases how methods like Label Distribution Aware Marginal Loss, hierarchical classification, and FixMatch address the three challenges. We also present a new approach to tackle the annotation uncertainty challenge using a soft-fine pseudo-label methodology. Finally, we aim with this chapter to increase awareness in the ecological remote sensing community of these challenges and bridge the gap between ecological applications and state-of-the-art computer science, thereby opening new doors to future research.

## Acknowledgments

We thank Brian Lubinksi and Dave Fronczak for their assistance with the data collection and annotation of data used in the study. The findings and conclusions in this chapter are those of the author(s) and do not necessarily represent the views of the U.S. Fish and Wildlife Service.

# Introduction

Aerial remote sensing technologies are being increasingly used to monitor and survey wildlife populations [142, 133]. They provide non-invasive tools for detecting, classifying, and assessing the abundance of target sites [80]. Traditional wildlife aerial surveys employ human observers to conduct visual counts, often from low-flying aircraft. While these methods can be efficient in surveying large geographic regions, visual observations from low-flying aircraft are risky (i.e., observation personnel can face life-threatening risks because of low-flying altitudes) [107] and are subject to various observer biases such as count bias [100, 34]. In contrast, aerial remote sensing is a relatively safer alternative that allows flying at a higher altitude, and the method offers the potential for consistent and reproducible population survey with the addition of an accurate geo-referenced digital format. In addition, aerial imagery surveys conducted at higher altitudes may also reduce animal disturbance [107]. The major disadvantage of using remote sensing for aerial surveys is that covering large areas can generate hundreds of thousands of images, thus hundreds of terabytes of data. Therefore, manually processing remote sensing aerial survey data is time consuming and prohibitively expensive for many researchers and natural resource agencies [15].

Ecologists are increasingly looking to cutting-edge artificial intelligence (AI) methodology, such as deep learning and computer vision technologies, to mitigate the need for extensive manual processing of digital aerial imagery and to improve monitoring efficiency. For example, deep learning has now been applied to aid in digital aerial surveys conducted with various sensor systems, including RGB [50, 72], thermal [21], and other sensor systems [142]. However, real-world digital aerial imagery applications of AI methods must address several challenges limiting recognition performance. These include: the imbalanced distribution challenge—extremely imbalanced data distributions that generally lead to poor recognition performance; the annotation uncertainty in categorization challenge—uncertainty in annotation caused by various reasons such as varying image resolutions of avian individuals; and the dataset discrepancy challenge—images collected from different study sites (i.e., geographies) that have different characteristics and classes.

To examine these challenges in detail, we use a case study of two real-world digital aerial survey datasets of waterbird species: one collected from the Atlantic Ocean near Cape Cod, Massachusetts, and the other from Lake Michigan near Manitowoc, Michigan, USA. We also present solutions, accompanied by brief literature reviews for each challenge, focusing on how the computer science community has previously addressed these types of challenges. We aim to increase awareness of these challenges within the ecological community, clarify the factors affecting AI recognition performance, demonstrate the flexibility of deep learning methods, and promote future research in AI and digital aerial surveys.

## Avian recognition

Aerial images of birds may include a few or many individuals depending on resources being used by those birds and flocking behavior displayed by species-specific bird groups. Thus implementation of AI methods for identifying the species consists of two distinct tasks (Figure 2.1): 1.) identifying and cropping out (also referred to as detecting and bounding) all the individuals in the image; and 2.) recognizing species and type (e.g., male or female, sub-adult or adult).

AI cropping methods of birds in aerial images remains to be better developed, although AI counting of the number of individuals in large aggregations exist [29], automated methods for delineating trees in aerial images of forests is quite well developed [25], and deep learning methods have been developed for segmentation of various types of geographical objects (e.g., land cover and land use types) from aerial images [139].

In this chapter, we focus on task 2 because task 1 has been addressed by studies like [50]. In other words, we consider only sets of data that consists of images of individuals that have already been cropped out either manually or through the application of AI procedures. The task at hand then is to build an AI model that automatically recognizes (i.e., classifies) avian species from aerial image segments cropped to include only one individual, often at relatively low levels of resolution.

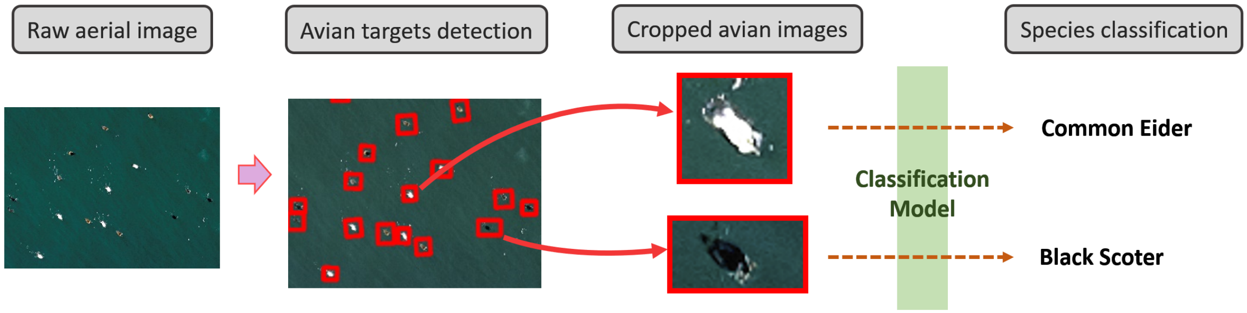


Figure 2.1: **This example shows how a raw aerial image is processed into final species classification in our case study.** Once the raw aerial images are collected, potential objects in the raw images are detected and bounded with boxes either manually or by automatic detection tools [50]. We used manual bounding boxes from human annotators in our case study. Once the potential objects were cropped around the bounding boxes, our task was to build a deep learning classification model to recognize the actual avian species from these cropped images.

## Dataset

For our case study, we used an aerial imagery dataset collected from two study sites over bodies of water: the Atlantic Ocean near Cape Cod, Massachusetts and Lake Michigan near Manitowoc, Michigan, USA. After data collection, wildlife experts manually annotated and cropped images of individual birds (i.e., targets; Figure 2.1). These images were then passed to a classification algorithm for species classification [72, 40]. The 10,682 individuals identified in the Cape Cod dataset and 236 identified in the Lake Michigan dataset were annotated by experts into the six different classes illustrated in Figures 2.2 and 2.3:

1. `Unknown Scoters` (scoter individuals that human annotators could not distinguish to the species level)

2. `Black Scoter` (*M. americana*)

3. `White-winged Scoter` (*M. deglandi*)

4. `Common Eider` (*Somateria mollissima*)
5. `Long-tailed Duck` (*Clangula hyemalis*)
6. `Non-target Species` (all other avian or non-avian individuals not belonging to the previous classes).

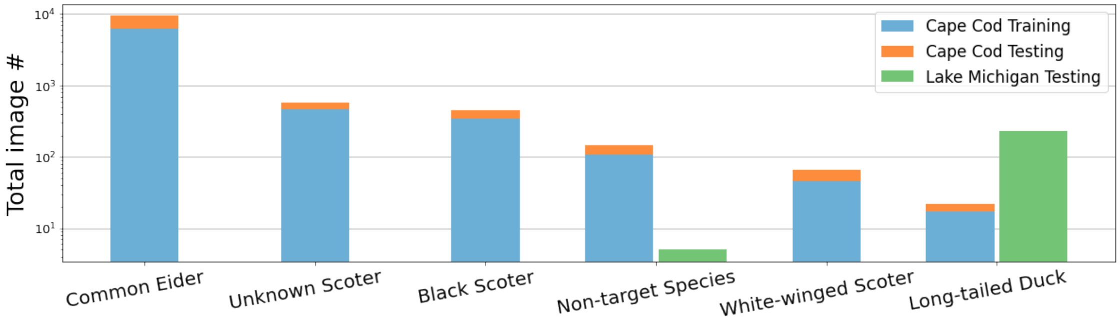


Figure 2.2: **The distribution of our dataset is imbalanced.** The classes are sorted by the sample sizes of each class in Cape Cod. The blue, orange and green colors represent training and testing images from Cape Cod and testing images from Lake Michigan, respectively. In Cape Cod, the largest class, `Common Eider`, has over 6,246 training images, while the smallest class, `Long-tailed Duck`, only has 17 training images. In other words, the imbalance ratio of the Cape Cod dataset is 367:1. Lake Michigan dataset only has two classes, `Long-tailed Duck` and `Non-target Species`, and it is also imbalanced in terms of class sizes. `Long-tailed Duck` from Lake Michigan has 231 images, while there are only 5 images for `Non-target Species`.

Our model was mainly trained on the Cape Cod dataset. Lake Michigan data were used to evaluate the model's generalization ability. In other words, we used Lake Michigan data to examine whether the model trained on the Cape Cod dataset could generalize well on the Lake Michigan dataset. The details of data pre-processing for the experiments are reported in the Data Section of the Appendix.

## Challenges of avian recognition in aerial imagery

### Training with a standard deep learning classification model

We started by applying a standard six-class classification model (i.e., the fundamental classification model without any additional components designed for tasks other than classification) to our Cape Cod dataset because there are six classes (i.e., we treated the `Unknown Scoter` and the other two scoter classes as three separate classes). The model we used was ResNet-50 [46], a common deep learning Convolutional Neural Network (CNN). The test results from this model are reported in Table 2.1. The implementation and hyperparameter tuning details are in the Appendix Methods section.

Table 2.1 shows that in the Cape Cod test set, except for the largest class (i.e., most frequently observed class), `Common Eider`, which had a 99.0% test accuracy, the remaining classes did not produce accurate recognition performance with the standard classification model. Specifically, the two smallest classes (i.e., least observed classes), `Long-tailed Duck`

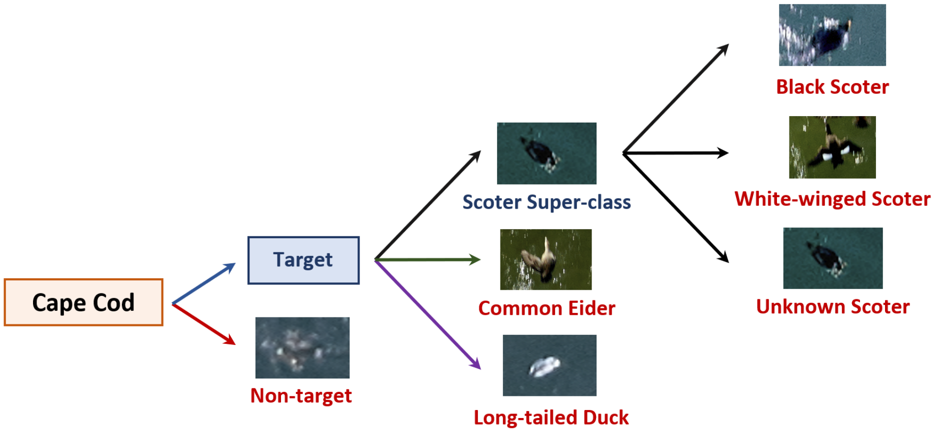


Figure 2.3: **The 6 classes in the Cape Cod dataset have a hierarchical relationship.** Firstly `Non-target` plus 3 target classes—`Scoter Super-class`, `Common Eider`, `Long-tailed Duck`; and `Scoter Super-class` could further be categorized/annotated as `Black Scoter`, `White-winged Scoter`, and `Unknown Scoter`. The reason not all `Scoter Super-class` images could be further categorized to the species level was related to image resolution in our dataset—low resolution images posed significant difficulties for human annotators to make accurate annotations of whether some images were `Black Scoter` or `White-winged Scoter`. Specifically, the average image dimension of `Unknown Scoter` was 56×61, while the average image dimensions of `Black Scoter` and `White-winged Scoter` were 100×107 and 96×103, respectively. `Unknown Scoter` can be considered a coarse annotation of `Black Scoter` and `White-winged Scoter` because it contains images of either one of the two scoter classes but without species-level annotations. On the other hand, we note that the `Non-target Species` class includes images that do not belong to the other five classes. In other words, images between `Target` and `Non-target Species` are mutually exclusive in our datasets. Class labels used for classification are colored in red.

Table 2.1: **The standard classification model trained on the Cape Cod training set performed poorly on the Cape Cod and the Lake Michigan test sets.**

| Species | Cape Cod | | Lake Michigan | Test accuracy (%) | |
|---|---|---|---|---|---|
| | Train # | Test # | Test # | Cape Cod | Lake Michigan |
| `Unknown Scoter` | 466 | 114 | - | 69.3 | - |
| `Black Scoter` | 341 | 108 | - | 55.6 | - |
| `White-winged Scoter` | 45 | 21 | - | 9.5 | - |
| `Common Eider` | 6,246 | 3,172 | - | 99.0 | - |
| `Long-tailed Duck` | 17 | 5 | 231 | 0.0 | 0.0 |
| `Non-target Species` | 108 | 38 | 5 | 18.4 | 20.0 |
| Average accuracy (%) | | | | 41.9 | 10.0 |

and `White-winged Scoter`, had only 0.0% and 9.5% test accuracy. This performance inconsistency is negatively related to the training size of each class. In other words, the fewer training images a class had, the less accurate the model was. We also tested our model on the Lake Michigan data, and the performance were also poor. These results indicate that directly applying a standard classification model on our avian datasets is insufficient to produce good recognition performance. Next, we discuss the causes of this performance inconsistency in the context of the imbalanced distribution, annotation uncertainty in categorization, and the dataset discrepancies challenges.

**Challenge 1:** ***Avian imagery data are naturally imbalanced***

Data collected during multi-species surveys tend to have an imbalanced species distribution (i.e., long-tailed distribution) because of the natural composition of animal communities [93]. Several dominant species are often observed along with many infrequent species that are sparsely represented in datasets. As illustrated in Figure 2.2, in the Cape Cod dataset, the largest class had 6,246 training images, while the smallest class only had 17 training images. This training data distribution imbalance leads to a significant recognition performance inconsistency. The fewer training images of a particular species that the model has, the lower the accuracy for that species. In our experiment, the performance was particularly poor for species with smaller training datasets, such as `Long-tailed Duck` (17 training images) and `White-winged Scoter` (45 training images), which only had 0.0% and 9.5% test accuracy, respectively (Table 2.1). However, `Common Eider`, the largest class in the dataset (6,246 images), had a 99.0% test accuracy.

**Challenge 2:** ***Annotation uncertainty in categorization***

Sometimes aerial data can be collected from aircraft at various distances from the ground surface, resulting in varying spatial resolutions as measured by Ground Sampling Distances (GSD; i.e., the ground distance between the centers of neighboring image pixels), and thus, the same species may appear at different image resolutions within a dataset. For example, in the Cape Cod dataset, the average image dimension of `Unknown Scoter` images was 56×61, while the average image dimention of `Black Scoter` and `White-winged Scoter` were 100×107 and 96×103, respectively. In other words, `Unknown Scoter` images contain 3-4 times fewer pixels on average. Low resolution images increase human annotators' difficulties in making accurate classifications resulting in a coarse annotation rather than individual species annotation of scoter images. `Unknown Scoter` is one example of coarsely annotated class.

Directly incorporating this coarsely annotated class as an independent class confused the classification models significantly because the model was forced to distinguish similar-looking avian objects as different classes. In other words, since `Unknown Scoter` contains images that can be either `Black Scoter` or `White-winged Scoter`, although most `Unknown Scoter` images were difficult for human annotators to determine the exact scoter species, they still share similar visual features with `Black Scoter` and `White-winged Scoter`. For example, Figure 2.4 shows that even though `Unknown Scoter` and `Black Scoter` had relatively sufficient training images (466 and 341 training images, respectively), 40.7% `Black Scoter` images were misclassified as `Unknown Scoter`, and 23.7% `Unknown Scoter` images were misclassified as `Black Scoter`. In addition, 61.9% `White-winged Scoter` were misclassied as `Unknown Scoter`.

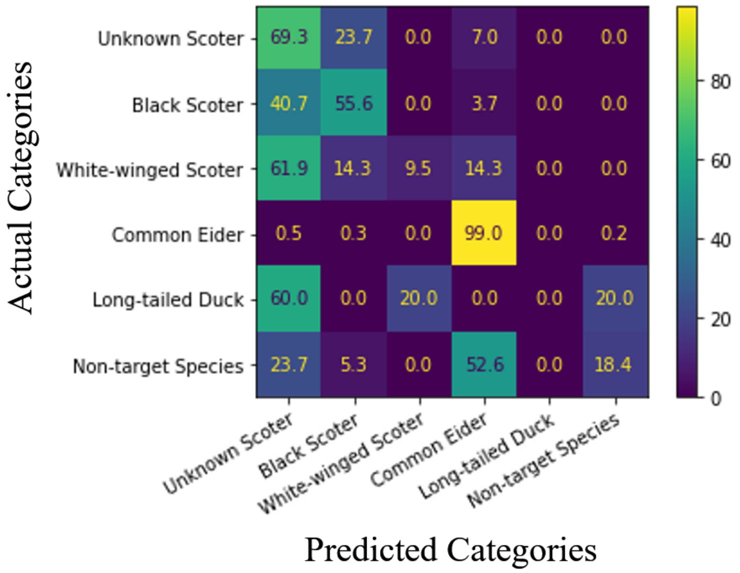


Figure 2.4: **On the Cape Cod test set, the model generally performed poorly because of the imbalanced data distribution and there exist substantial recognition confusion among the three scoter classes.** This is the confusion matrix of the standard model we trained on the Cape Cod dataset. The rows of the matrix represent the actual classes, the columns represent the predicted classes, and the values represent the percentages of predictions of each class. The matrix shows that our model did not perform well on smaller classes like `Long-tailed Duck` and `White-winged Scoter`. In addition, as shown in the top left corner, the three scoter classes were confused with each other.

### Challenge 3: *Dataset discrepancies often arise among different study sites*

Besides the imbalanced distribution and annotation uncertainty challenges, in practice, ecological monitoring projects often expand over time [121]. New monitoring and study sites are added, leading to discrepancies among datasets in lighting conditions, background environment, atmospheric conditions, image capturing distances, and animal species compositions. For example, in our case study, Cape Cod and Lake Michigan datasets have different GSDs, which result in different image resolutions and appearances of avian individuals from the same species (Figure 2.5). The `Long-tailed Duck` images have 3-4 times the resolution (number of pixels) of Lake Michigan images and thus contain more visual details and features. As a result, a classification model trained on low resolution images (Cape Cod) may perform poorly on images with higher resolution (Lake Michigan). For example, the standard model trained on Cape Cod dataset only had a 10.0% test performance on the Lake Michigan dataset (Table 2.1). We demonstrate in the following Methods and results Section that this poor performance did not only come from imbalanced distribution.

In addition to image appearance discrepancies in datasets from different study sites, expanding surveys or monitoring programs can also change the composition of animal species recorded [58]. For example, as data collections continue over time, previously undetected species may also be encountered [96] (e.g., less frequent species [93], recolonizing species [81],

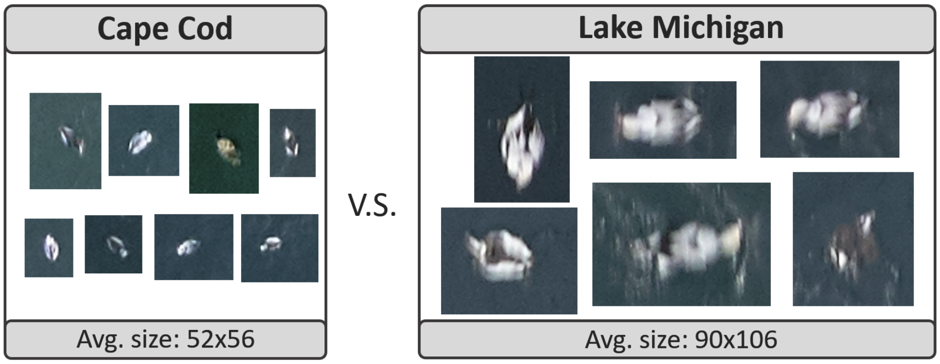


Figure 2.5: `Long-tailed Duck` **images from Lake Michigan are 3-4 times larger than those from Cape Cod.**

reintroduced animals [130, 101], or invasive species that are harmful to the ecosystem [20, 14]). When novel species are introduced, our standard classification model is no longer effective because conventional AI methods require datasets to have fixed numbers of classes [5]. Therefore, novel species are typically unrecognizable.

# Methods and results

In this section, we provide brief literature reviews of how the computer science community addresses the challenges mentioned in the previous section and present solutions to each challenge.

## Solutions for the imbalanced distribution challenge

Imbalanced recognition and long-tailed recognition are areas of machine learning and computer vision research that address imbalanced classification problems [74, 12, 143]. Common methods include:

1. **Training data resampling:** artificially balancing training datasets by either sampling more images from smaller classes (i.e., up-sampling) or sampling fewer images from larger classes (i.e., down-sampling) [43].

2. **Training loss re-weighting:** assigning different weights (i.e., training focus) to the training loss functions based on the numbers of training images of each class such that the model can focus more on smaller classes [69, 23, 12].

3. **Knowledge transfer:** transferring information (such as semantic and visual knowledge) from larger classes to smaller classes such that data from smaller classes can provide more distinguishable information for better classification performance [74, 56, 154]. For example, Liu et al. [74] proposed a method that utilized information learned from larger classes to enhance the distinguishability of smaller classes.

4. **Model ensemble:** ensembling outputs from multiple expert models (i.e., sub-models) for optimal performance [154, 143, 11].

Specifically, Routing Diverse Distribution-Aware Experts (RIDE) and Ally Complementary Experts (ACE) [143, 11] are the two state-of-the-art methods using modern multi-expert ensembling methods and have the highest performance on standard imbalanced benchmarks (such as ImageNet-LT [74]). However, the tuning of multiple expert models involves many hyperparameters and these methods tend to work better with a larger number of classes (e.g., species).

For a relatively smaller scale classification task (e.g., six-class classification in our case study), we used a light-weighted loss re-weighting method called Label Distribution Aware Marginal loss (LDAM) [12] to address the imbalanced data distribution. Generally speaking, LDAM calculates class-specific margins (i.e., classification certainty) based on class sample sizes of each class. In other words, the fewer training samples a class has, the larger the class-specific margin is, and vice versa. Details of the LDAM we used are reported in the Method Section of the Appendix.

The classification model with a imbalanced component (LDAM in our experiments) substantially improved the recognition performance on the Cape Cod dataset over the standard classification model (Table 2.2). The average class accuracy improved from 41.9% to 75.8%. The largest gain came from the two smallest classes, `Long-tailed Duck` and `White-winged Scoter`, from 0.0% to 100.0% and 9.5% to 81.0%, respectively. Despite the improvement in the less abundant classes, the performance of `Common Eider` dropped by 7.1%, which is a common phenomenon of imbalanced methods where the performance of large classes is sacrificed [74, 143].

Table 2.2: **The imbalanced model substantially improved the test performance from the standard model on the Cape Cod test set.**

| | | | Test accuracy (%) | |
|---|---|---|---|---|
| **Species** | **Train #** | **Test #** | **Standard** | **Imbalanced** |
| `Unknown Scoter` | 466 | 114 | 69.3 | 41.2 |
| `Black Scoter` | 341 | 108 | 55.6 | 59.3 |
| `White-winged Scoter` | 45 | 21 | 9.5 | **81.0** |
| `Common Eider` | 6,246 | 3,172 | 99.0 | **91.9** |
| `Long-tailed Duck` | 17 | 5 | 0.0 | **100.0** |
| `Non-target Species` | 108 | 38 | 18.4 | 81.6 |
| **Average accuracy (%)** | | | 41.9 | **75.8** |

The confusion matrices (Figure 2.6) show that the imbalanced model cleared most of the confusion in `Long-tailed Duck` and `White-winged Scoter` because LDAM assigned larger margins to these classes with limited training samples. However, the imbalanced model still struggled to perform well on `Unknown Scoter` and `Black Scoter`, with only 41.2% and 59.3% test accuracy, respectively. From Figure 2.6 column (b), it is clear that the confusion within the three scoter classes was still significant. For example, the imbalanced model misclassified 44% of `Unknown Scoter` as either `Black Scoter` or `White-winged Scoter`. Meanwhile, 31% `Black Scoter` and 19% `White-winged Scoter` were misclassified as `Unknown Scoter`.

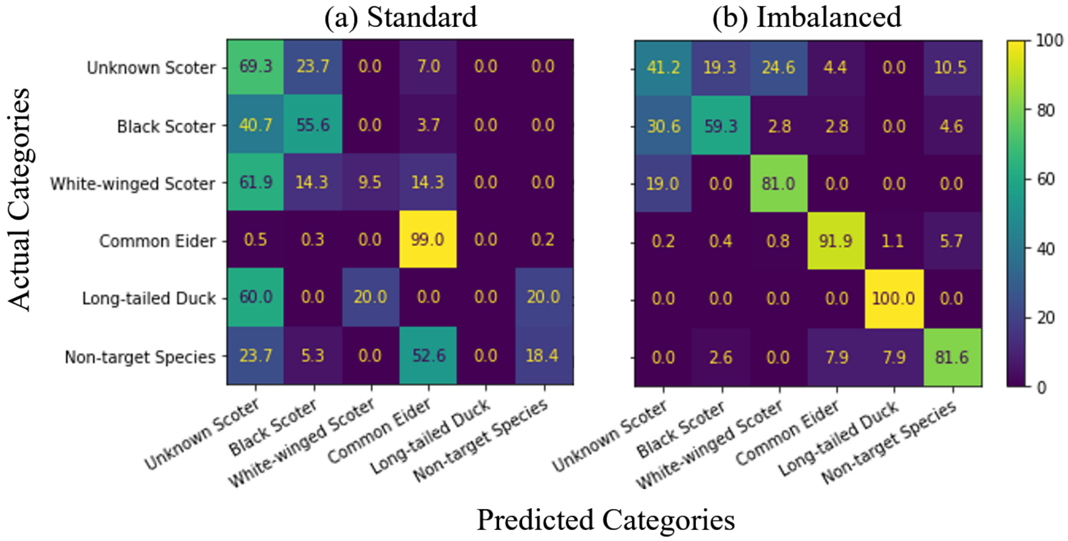


Figure 2.6: **On the Cape Cod test set, the imbalanced model cleared the confusion in smaller classes, but the confusion among the scoter classes persisted.** Specifically, compared to the standard model, the imbalanced model cleared most of the confusion in the lower right parts of the matrices (especially for `Long-tailed Duck`, which had limited training images), but the confusion in the top left corner (where the three scoter classes are located) was still significant. In other words, the use of the imbalanced component only helped the model to have better performance on classes with limited training samples, such as `Long-tailed Duck` and `White-winged Scoter`.

## Solutions for the annotation uncertainty in categorization challenge

Because `Unknown Scoter` introduced significant confusion to the model and the recognition of `Unknown Scoter` does not provide species-level information for downstream tasks like population modeling, it is more practical to exclude these coarsely annotated data from model training to eliminate the confusion. However, directly excluding coarsely annotated data is a sub-optimal solution because images with different resolutions can provide complementary information that ultimately improves the generalization abilities of classification models [68]. Since `Unknown Scoter` in our dataset is believed to be the class of relatively low resolution images of either `Black Scoter` or `White-winged Scoter`, these images may still provide information to improve model performance at the species level, especially when ground-truthed annotations (i.e., human annotations in this context) are limited. For example, though the imbalanced model vastly improved the test accuracy of `White-winged Scoter` from 9.5% to 81.0% (Table 2.2), there is still space for performance gain by exploiting information contained in `Unknown Scoter`.

### Hierarchical classification

Classifications using hierarchical classification methods is one of the common options addressing uncertain and coarse annotations [28]. To demonstrate the effects of hierarchical classification methods, we split the training process into two stages. In the first stage, we merged Unknown Scoter, Black Scoter, and White-winged Scoter data into one single super-class, Scoter Super-class, and trained the classification model on 4 independent classes (Scoter Super-class, Common Eider, Long-tailed Duck, and Non-target Species). Then, we trained a separate classifier to classify only Black Scoter and White-winged Scoter in the second stage (i.e., the same classification model as in the first stage but for two-class classification). As a result, separate classifiers with grouped Scoter Super-class in the first stage removed considerable confusion among the scoter classes because the model did not have to distinguish images among the scoter classes (Table 2.3). In the first stage, the imbalanced model had a 94.2% test accuracy on the grouped Scoter Super-class. In addition, the imbalanced model in the second stage produced an average of 92.5% test accuracy on Black Scoter and White-winged Scoter, which was significantly improved compared to the six-class experiment (70.2% averaged accuracy on Black Scoter and White-winged Scoter; Table 2.2 under Imbalanced column).

Next, we combined the two-stage evaluations to examine the effective performance of Black Scoter and White-winged Scoter because practically, the evaluation performance of second-stage models depend on the performance of Scoter Super-class in the first-stage evaluation. We thus considered predictions of Black and White-winged Scoter positive only when the positive prediction was also positive in the first-stage Scoter Super-class predictions. As a result, when the evaluation was combined, the average test performance of the Black and White-winged Scoter significantly dropped from 77.3% to 56.3% using the standard model and from 92.5% to 51.5% using the imbalanced model (Table 2.3 under Combined columns).

Further, when the model had imbalanced components (see column Imbalanced under Combined in Table 2.3), the performance of White-winged Scoter was significantly worse than the separate evaluation (9.5% when combined versus 90.5% when separate). Because White-winged Scoter is a class with limited training samples, it did not benefit from the Imbalanced component when merged with Black Scoter and Unknown Scoter. In other words, because of the lack of training images, the model could not generalize on White-winged Scoter and misclassified most of the images in the first stage. Therefore, there were not enough images for the second-stage model to recognize, even though the second-stage model performed well separately.

Table 2.3: **On the Cape Cod test set, two-stage models performed well separately, but the combined effective performance was poor.**

| | | | | **Test accuracy (%)** | | | |
|---|---|---|---|---|---|---|---|
| | **Species** | **Train #** | **Test #** | **Separate** | | **Combined** | |
| | | | | **Standard** | **Imbalanced** | **Standard** | **Imbalanced** |
| **Stage 1** | Scoter Super-class | 852 | 242 | 93.8 | 94.2 | 93.8 | 94.2 |
| | Common Eider | 6,246 | 3,172 | 99.1 | 93.0 | 99.1 | 93.0 |
| | Long-tailed Duck | 17 | 5 | 0.0 | 100.0 | 0.0 | 100.0 |
| | Non-target Species | 108 | 38 | 23.7 | 68.4 | 23.7 | 68.4 |
| **Stage 2** | Black Scoter | 341 | 108 | 92.6 | 94.4 | 88.8 | 93.5 |
| | White-winged Scoter | 45 | 21 | 61.9 | 90.5 | 23.8 | 9.5 |
| **Average accuracy of** Black **and** White-winged Scoter**(%)** | | | | 77.3 | 92.5 | **56.3** | **51.5** |

In addition, training with multiple stages can quickly become a scaling and model management problem if the dataset has multiple super-classes. Each super-class requires an independent second-stage model and training process. As the number of super-classes increases, the number of models grows as well, such that the overall training time and model management efforts are significantly increased.

### A novel solution: Soft-fine Pseudo-labels

A better solution is to exploit additional information from coarsely annotated `Unknown Scoter` images without including it as an independent class and keep the imbalanced component effective on `White-winged Scoter` at the same time. Therefore, we applied a novel solution called *Soft-fine Pseudo-Labels* (SPL) to address the coarse/uncertain annotation problem that relied only on one stage of training. The method is derived from pseudo-label techniques, a set of techniques in machine learning that utilize model predictions (i.e., pseudo-labels) to improve the generalization ability of machine learning models [66, 117]. Common approaches in this area include:

1. **Pseudo-labels with confidence metrics:** generating pseudo-labels with arbitrary confidence metrics, where only high confidence predictions are accepted as pseudo-labels [66] to train the models.

2. **Consistency regularization:** models are trained to produce consistent outputs of the same input (image in this context) with different perturbations (e.g., data augmentation [35] and stochastic regularization like content dropout [120]). In other words, model outputs of the same input should be the same regardless of the perturbation such that the models can produce higher prediction confidence and thus pseudo-labels with better quality [117, 147, 146].

Figure 2.7 illustrates the differences of treating the three scoter classes using our SPL approach compared to standard and two-stage models. Unlike conventional pseudo-label approaches that generate pseudo-labels for all possible classes, in our approach, we only use `Unknown Scoter` images to generate finer-grained `Black` and `White-winged Scoter` pseudo-labels that are beneficial to model generalization. Specifically, we first normalized the outputs of the classification model (5 dimension vectors) with a Softmax function [48]. Then we normalized the values that represent `Black Scoter` and `White-winged Scoter` to 1 and set the other three values to 0 (Figure 2.8). We used these normalized Softmax values as our soft-fine labels on `Unknown Scoter` images with an Averaged Binary Cross-entropy (ABCE) loss, a loss function traditionally used for samples with multiple co-occurring labels [132].

Our SPL approach forces the model to distinguish between scoter versus non-scoter images because the training signals (i.e., the generated soft-fine pseudo-labels) have zeros on the dimensions that represent non-scoter classes. In addition, since the generated soft-fine pseudo-labels had co-occurring labels for both `Black Scoter` and `White-winged Scoter` that were treated independently by ABCE loss, the confusion between these two scoter classes was also suppressed. The implementation details can be found in the Appendix Methods section, and the results for our case study are reported in Table 2.4.

If `Unknown Scoter` is excluded, the task becomes a five-class classification problem. In our experiment, the average class accuracy of fully excluding `Unknown Scoter` from training and testing improved from 41.9% to 53.2% on the standard model and 75.8% to 89.5% on

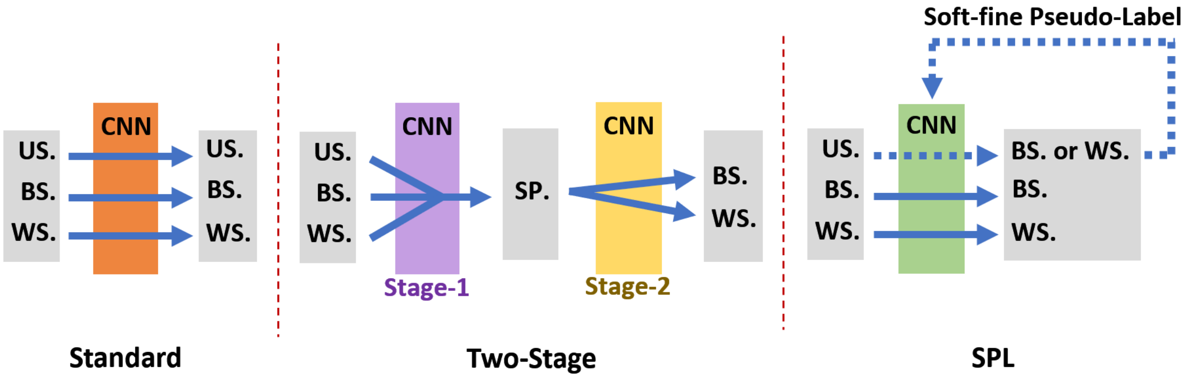


Figure 2.7: **We graphically depict how the three scoter classes can be classified using different approaches.** Under the standard approach, `Unknown Scoter` is treated as an independent class. Under the two-stage setting, the three scoter classes are firstly grouped into one super-class, `Scoter Super-class`, and then a separate model is trained solely for `Black` and `White-winged Scoter` classification. With our SPL approach, coarsely annotated `Unknown Scoter` images are converted to finer-grained pseudo-labels and used to improve model generalization. US. is `Unknown Scoter`. BS. is `Black Scoter`. WS. is `White-winged Scoter`. SP. is `Scoter Super-class`.

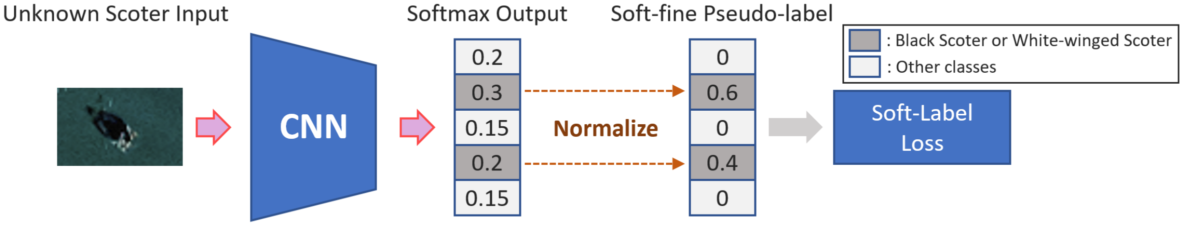


Figure 2.8: **A diagram of our SPL approach to solving the annotation uncertainty challenge using a novel soft-fine pseudo-labeling method.** The soft-fine labels are generated by normalizing the Softmax outputs of `Unknown Scoter` images. To generate the soft-fine pseudo-labels for coarsely annotated `Unknown Scoter` images, first, we normalized the two values representing `Black Scoter` and `White-winged Scoter` from the Softmax outputs to 1 and set the other values to zero. Then, the new vectors were used as the pseudo-labels with soft supervision (i.e., the supervision values are less than 1) on either `Black Scoter` or `White-winged Scoter`. With this approach, the supervision from `Unknown Scoter` is not as strong as independent classes but still relevant to force the model to recognize the images as scoters with higher probabilities than the other classes. Further, this framework does not rely on multiple stages of training and class merging, such that the imbalanced model can still be effective on `White-winged Scoter`.

the imbalanced model compared to the six-class classification results because there was no confusion from `Unknown Scoter` (Tables 2.2 and 2.4). On the other hand, incorporating `Unknown Scoter` for complementary information with our SPL approach further improved the test accuracy of `White-winged Scoter` from 85.7% to 90.5% compared to the imbalanced model, which did not use `Unknown Scoter` data during training (column Imbalanced + SPL in Table 2.4). In addition, performance on classes other than the two scoter classes was also improved, especially `Non-target Species`, which increased from 78.9% to 81.6% compared to the imbalanced model. These improvements indicate that the use of SPL with coarsely annotated data not only relieved the confusion among the scoter classes (Black Scoter and White-winged Scoter) but also among other classes (Figure 2.9). However, in hierarchical classification, the performance of other classes was almost the same, if not worse, compared to the six-class classification results (Tables 2.2 and 2.3). Further, the only additional components to the imbalanced model are the SPL normalization and ABCE loss, making this approach scalable without requiring multiple stages of training.

Table 2.4: **On the Cape Cod test set, our soft-fine pseudo-labeling (SPL) approach improved the performance of White-winged Scoter from the imbalanced model by exploiting Unknown Scoter.**

| | | | Test accuracy (%) | | |
|---|---|---|---|---|---|
| **Species** | **Train #** | **Test #** | **Standard** | **Imbalanced** | **Imbalanced + SPL (Ours)** |
| Black Scoter | 341 | 108 | 96.3 | 91.7 | 89.8 |
| White-winged Scoter | 45 | 21 | 33.3 | 85.7 | **90.5** |
| Common Eider | 6,246 | 3,172 | 99.4 | 91.2 | 91.5 |
| Long-tailed Duck | 17 | 5 | 0.0 | 100.0 | 100.0 |
| Non-target Species | 108 | 38 | 36.8 | 78.9 | **81.6** |
| **Average accuracy of the two scoter classes (%)** | | | 64.8 | 88.7 | **90.1** |
| **Average accuracy (%)** | | | 53.2 | 89.5 | **90.7** |

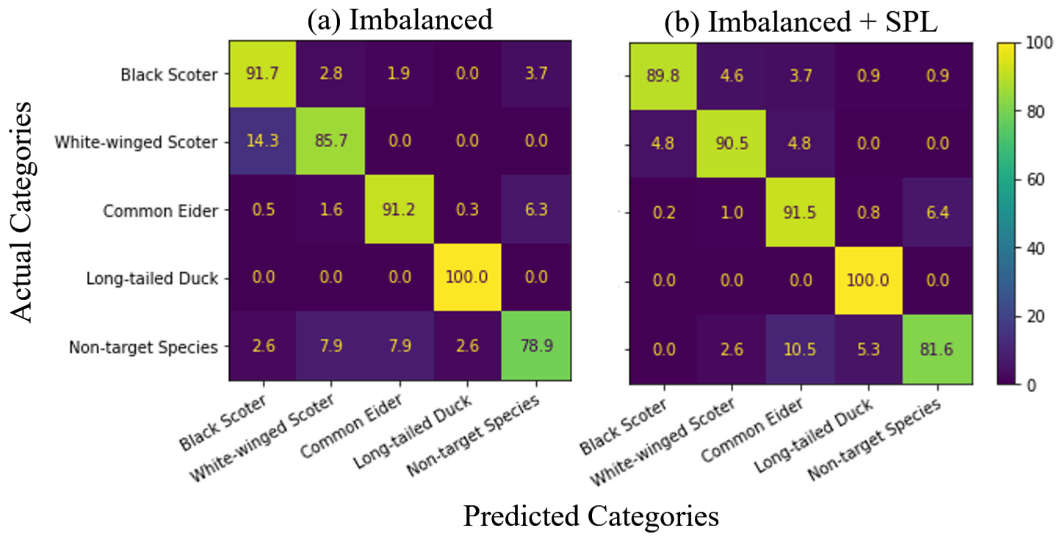


Figure 2.9: **Our SPL approach further reduced the confusion within Black Scoter and White-winged Scoter from the imbalanced model within the Cape Cod test set.** Comparing the top-left corners of the confusion matrices, most of the misclassified White-winged Scoter images in the imbalanced model were correctly classified with our SPL approach. However, additional uncertainty can still be introduced to the model because SPL did not provide ground-truthed supervision (i.e., only model-produced pseudo-labels were provided), which can be the reason of the slightly degraded performance for Black Scoter, from 91.7% to 89.8%.

Despite the scalability and the exclusion of coarse annotation confusion, the proposed SPL can sacrifice some performance in the two scoter classes (Black Scoter and White-winged Scoter) compared to hierarchical classification. For example, the effective test accuracy of Black Scoter was 93.5% using two-stage training, while it was only 89.8% using our soft-fine label approach (Tables 2.3 and 2.4). The uncertainty during SPL training was the leading cause of this performance drop because the model was trained to identify all species at once with pseudo-labels (i.e., labels that are not ground-truthed).

### Unknown Scoter evaluation

On the other hand, in our five-class classification experiments, we only focused on the test performance of samples with finer-grained annotations. With our SPL approach, although we were able to exploit Unknown Scoter data during training, they were also excluded from testing because of the lack of finer-grained annotations. In addition, since some of the Unknown Scoter images are too blurry to be recognized, directly applying classification models to provide fine-grained predictions can be an issue as well (Figure 2.10). How to efficiently address these blurry images during test time is one of the future research directions.

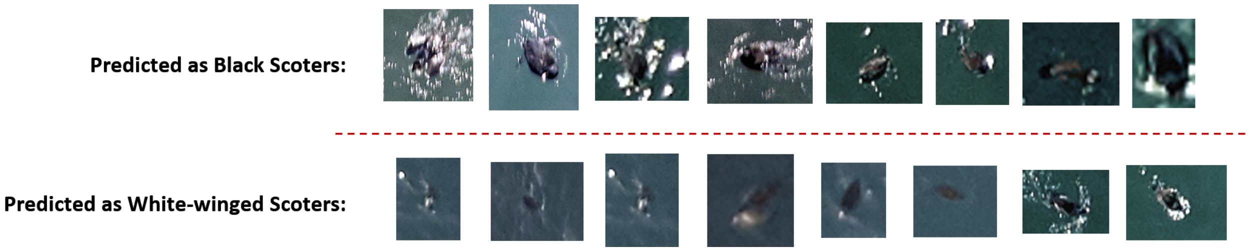

Predicted as Black Scoters:
Predicted as White-winged Scoters:

Figure 2.10: **Examples of Unknown Scoter test images predicted as Black Scoter and White-winged Scoter.** Although we can apply our model to assign fine-grained predictions to Unknown Scoter test images of Cape Cod, it is difficult to evaluate the performance because of the lack of ground-truthed annotation and low-resolution.

## Solutions for the dataset discrepancy challenge

### Visual discrepancies

Next, we tested the performance of our SPL model trained from the Cape Cod dataset on the Lake Michigan dataset. Because of the visual discrepancies between the Cape Cod and Lake Michigan datasets, the average accuracy of the two classes dropped substantially from 90.8% to 65.1% (Tables 2.4 and 2.5). Only 50.2% of the Long-tailed Ducks in Lake Michigan data were correctly classified when the test accuracy on the Cape Code dataset was 100.0% (Table 2.5 under Imbalanced + SPL column). These results also show that after the imbalanced distribution challenge was addressed, the model still did not perform well on Long-tailed Duck from Lake Michigan.

One of the most common approaches to address the challenge of incorporating data from new study sites is to fine-tune existing models (i.e., transfer learning) with new annotations [148]. In our example, we need to provide sufficient annotated Lake Michigan data to fine-tune our Cape Cod model such that the model can recognize targets from both study sites. Although the total image number of the Lake Michigan dataset is relatively small (236 images in our case study) and thus easy to annotate, the effort and resources needed for human annotation on larger datasets are not trivial.

Two machine learning techniques that can help an AI classification model adapt to new sets of data that look different without human annotations are the following:

1. **Domain adaptation:** this technique adapts models trained from one domain (study sites in this context) to other domains either with or without annotations [106, 136, 92]. Although they can be recognized as the same classes by humans, images from different domains tend to have different distributions in terms of color, texture, and visual appearances. These differences result in distribution discrepancies of the learned feature/latent vectors at the end of CNN models. Thus, *distribution confusion* is the most commonly adopted domain adaptation technique. The technique *confuses* the feature vector distributions of each domain (usually without class annotations) such that the models cannot distinguish which domain the feature vectors come from and learn to utilize more fundamental information (e.g., structural similarities) to make recognition [134, 49, 75]. While domain adaptation approaches with distribution confusion may perform better than most other methods, they usually require complicated distribution matching and confusion techniques. For example, the state-of-the-art Open Compound Domain Adaptation (OCDA) [75] method requires four stages of training and tuning and largely relies on considerable training data, which can be too complicated for smaller datasets with a limited number of classes and training data.

2. `Semi-supervised learning:` This technique is an alternative option and is usually more straightforward in terms of implementation [66, 117]. Semi-supervised learning uses unannotated data to improve the generalization ability of AI models, usually through the generation of pseudo-labels [155]. Intrinsically, similar to the mechanisms of advanced domain adaptation approaches with distribution confusion, semi-supervised learning also expands the feature vector distribution by learning from unannotated data [155]. In practice, when data are collected from new study sites, they are treated as unannotated data, and pseudo-labels are then generated for fine-tuning existing models.

Here, we explored how a relatively easy-to-implement semi-supervised learning method, FixMatch [117], adapted our Cape Cod model to Lake Michigan images. FixMatch is lightweight because the only extra component required by FixMatch is a two-branch training data augmentation procedure. It can be easily plugged into our SPL model and other existing AI models. The details of this method are provided in the Appendix Methods section.

In Table 2.5 we report results of applying FixMatch as the adaptation component to fine-tuning the Cape Cod model on the Lake Michigan data. Although FixMatch was not initially designed for domain adaptation (i.e., only for semi-supervised learning tasks), it still substantially improved the classification accuracy on the Lake Michigan Lake dataset without any annotations. Compared to our SPL approach without the adaptation component, the class averaged accuracy went from 65.1% to 80.5%. Most of the improvements came from `Long-tailed Duck`, which increased its accuracy from 50.2% to 80.9%.

Table 2.5: **With FixMatch as the adaptation component, our model trained on the Cape Cod dataset performed substantially better than methods without the adaptation component.**

| | | Test accuracy (%) | | |
|---|---|---|---|---|
| **Species** | **Test #** | **Standard** | **Imbalanced + SPL** | **Imbalanced + SPL + Adaptation** |
| `Long-tailed Duck` | 231 | 0.0 | 50.2 | **80.9** |
| `Non-target Species` | 5 | 20.0 | 80.0 | **80.0** |
| **Average accuracy (%)** | | 10.0 | 65.1 | **80.5** |

### Novel species

When novel species are introduced, domain adaptation and semi-supervised learning methods are no longer effective because conventional AI recognition methods require datasets to have fixed numbers of classes [5]. Therefore, novel species are typically unrecognizable. Similar to adapting models to new domains, model fine-tuning through transfer learning with annotated data is also one of the most widely adopted methods to expand the models' recognition capacity [148]. However, since it is uncertain which individuals in the newly collected datasets are of novel species, a complete annotation (i.e., a considerable amount of human effort) is necessary for model fine-tuning.

In such circumstances, improving the efficiency of human annotation becomes a challenge. Ideally, it is possible to automatically identify all the images of novel species, and human effort can focus solely on these images rather than all the newly collected data. Out-of-distribution detection (OOD) [108, 30] is one of the related research areas in machine learning that attempts to discover novel samples during test time.

Modern OOD approaches for deep learning usually apply prediction confidence calibration to separate known and novel samples [67, 71]. In other words, since traditional SoftMax-based deep learning models are often overly confident (even on novel samples) [41], calibrating the confidence of sample predictions can be effective at separating known and novel samples. Common approaches include:

1. **Output smoothing:** smoothing the model outputs (e.g., Softmax output smoothing) to reduce the overconfidence of model predictions such that it is easier to find an effective prediction confidence threshold that separates known versus novel samples. [41, 67, 53, 126, 39, 71].

2. **Novel sample generation:** generating artificial novel samples (through data augmentation and generative models) to train AI models to produce lower confidence predictions on novel samples during testing [38, 47].

A more straightforward approach can be applied when there are non-target species in the dataset. In most real-world datasets, especially aerial imagery of small-bodied animals with uncertain human annotations, there are often instances of non-target animal species. When we treat these non-target instances as a single class, we can train AI models to classify target versus non-target animal species. Then all the images that are classified as non-target during test time can be sent to human experts for verification. Intrinsically, target versus non-target classification is an OOD technique. For example, Figure 2.3 shows that target versus non-target species are usually mutually exclusive, and a classifier can be learned between these two sets of classes. Thus, during test time, AI models are very likely to classify images of novel species as non-target species.

In comparing the methods listed in Table 2.5, we set an independent class in both Cape Cod and Lake Michigan datasets for non-target species. In the Lake Michigan data (Table 2.5), our model successfully identified most of the non-target species (with 80.0% test accuracy), even though non-target species in Lake Michigan did not necessarily overlap with those in the Cape Cod dataset. However, when there are insufficient training data for non-target species, it can be difficult for classification models to generalize well on novel species, and thus, more advanced OOD methods may be necessary.

## Conclusion

In this chapter, we tackled three challenges of automatic avian recognition in aerial imagery datasets and how various methods can be applied to addressing these challenges. We evaluated how well existing and our novel SPL approach performed with respect to these three challenges using data from Cape Cod and Lake Michigan.

First, we demonstrated that the classification performance of standard models is severely curtailed by an imbalance of the number of images of particular species. We showed that this imbalanced distribution challenge can be significantly mitigated by applying a light-weight imbalanced recognition method (LDAM), especially on classes with limited training samples like `Long-tailed Duck` and `White-winged Scoter`.

Second, we demonstrated that the classification performance of both standard one-stage and hierarchical classification methods was poor on data that included uncertainty in human annotations because of low resolution issues. This annotation uncertainty in categorization challenge results in some images being assigned to a coarse annotation (`Unknown Scoter`). We then demonstrated that classification performance could be much improved using our novel SPL approach that provides a link between coarse and fine-grained annotations. In particular, our approach generated soft-fine pseudo-labels from coarse `Unknown Scoter` annotations to improve the model's generalization ability on `Black Scoter` and `White-winged Scoter` classes. With our approach, we were able to exploit coarsely annotated data for better model generalization and keep the imbalanced component effective on `White-winged Scoter`, which had poor performance using hierarchical models because of the lack of training samples.

Third, we demonstrated that the test performance could be significantly improved using FixMatch when adapting models from data at one site to classifying data at another site. The dataset discrepancies challenge may often cause inconsistent classification performance. In our experiments, we attached FixMatch onto our SPL approach to address resolution discrepancies between datasets from Cape Cod and Lake Michigan and achieved better performance than baselines on the Lake Michigan data without additional annotations. We also experimented with the possibility of using a non-target class, `Non-target Species`, to detect novel species during testing. Our results show that the model could identify most of the `Non-target Species` images from the Lake Michigan dataset.

Although each solution we have discussed has its intrinsic limitations, these methods are often flexible and can be combined to accommodate specific requirements. For example, the imbalanced model with LDAM was combined with SPL and FixMatch to address imbalanced distribution, coarse annotations, and domain discrepancies. We have also demonstrated that existing methods can be easily adjusted for specific tasks. For example, our SPL approach is derived from pseudo-label and multi-label classification.

In addition, the solutions can be easily replaced by more advanced methods in the future if necessary. For example, when the number of training classes gets bigger, the imbalanced ratio within classes gets larger, and the data distribution get more long-tailed (i.e., a larger proportion of classes have limited training samples), LDAM can be replaced by methods like RIDE to produce optimal results. When the domain discrepancies among datasets get more complicated, such as multiple types of backgrounds, FixMatch can be replaced by domain adaptation methods like OCDA for unlimited possibilities of target domains.

On the other hand, some of the challenges we have listed are not specific to aerial avian recognition. For example, imbalanced and long-tailed distribution exists in ecological datasets derived from other sensor systems such as camera traps [83] and bio-acoustic monitors [19] because natural animal communities are imbalanced [93]. Through the examples presented here and the literature cited, we hope to demonstrate the flexibility of deep learning methods, open doors to the ecological community, and promote further research.

# Chapter 3

# Iterative Human and Automated Identification of Wildlife Images

Zhongqi Miao[1,2], Ziwei Liu[3,], Kaitlyn M. Gaynor[4], Meredith S. Palmer[5], Stella X. Yu[1,2], Wayne M. Getz[1,6,]

This chapter is published in Nature Machine Intelligence [83] and is included as a dissertation chapter with permission from co-authors.

[1]Dept. Env. Sci., Pol. & Manag., UC Berkeley, CA, United States
[2]International Comp. Sci. Inst., UC Berkeley, CA, United States
[3]School of Comp. Sci. & Eng., Nanyang Tech. Univ., Singapore
[4]Nat. Cent. for Eco. Ana. & Syn., UC Santa Barbara, CA, United States
[5]Dept. of Eco. & Evo. Bio., Princeton University, NJ, United States
[6]Sch. Math. Sci., Univ. KwaZulu-Natal, South Africa

## Abstract

Camera trapping is increasingly being used to monitor wildlife, but this technology typically requires extensive data annotation. Recently, deep learning has substantially advanced automatic wildlife recognition. However, current methods are hampered by the dependence on large static datasets, whereas wildlife data are intrinsically dynamic and involve long-tailed distributions. These drawbacks can be overcome through a hybrid combination of machine learning and human-in-the-loop. Our proposed iterative human and automated identification approach is capable of learning from wildlife imagery data with a long-tailed distribution. Additionally, it includes self-updating learning, which facilitates capturing the community dynamics of rapidly changing natural systems. Extensive experiments show that our approach can achieve ∼90% accuracy employing only ∼20% of the human annotations of existing approaches. Our synergistic collaboration of humans and machines transforms deep learning from a relatively inefficient post-annotation tool to a collaborative, ongoing annotation tool that vastly reduces the burden of human annotation and enables efficient and constant model updates.

# Acknowledgement

Thanks to T. Gu, A. Ke, H. Rosen, A. Wu, C. Jurgensen, E. Lai, M. Levy, and E. Silverberg for annotating the images used in this chapter and to everyone else involved in this project. Data collection was supported by J. Brashares and through grants to KMG from HHMI BioInteractive, the Rufford Foundation, Idea Wild, the Explorers Club, and the UC Berkeley Center for African Studies. We are grateful for the support of Gorongosa National Park, especially M. Stalmans, in permitting and facilitating this research. ZL is supported by NTU NAP. KMG is supported by Schmidt Science Fellows in partnership with the Rhodes Trust and the National Center for Ecological Analysis and Synthesis Director's Postdoctoral Fellowship. MSP is funded on the National Science Foundation grant, PRFB #1810586.

# Introduction

In our rapidly-changing world, continuous monitoring of natural systems is essential to understand and mitigate the impacts of human activity on ecological processes [121, 102, 7]. Recent technological innovations allow for the rapid collection of ecological data across vast spatial and temporal scales. However, the resulting information deluge creates a bottleneck for researchers who must process the data at management-relevant timescales [3]. Artificial Intelligence (AI) offers promising solutions for rapid and high-accuracy data processing [86, 82]. The dynamic nature of ecological systems, however, poses unique challenges when developing accurate algorithms [74, 75]. For example, in the last chapter, we have discussed three challenges of AI applications in real-world avian recognition. To overcome these hurdles, in this chapter, we showcase how the integration of limited human labor into the machine learning workflow can significantly increase both efficiency and accuracy of data processing.

## Long-term camera trapping

As discussed in Chapter 1, motion-activated remote cameras (henceforth camera traps) are popular non-invasive tools for monitoring terrestrial vertebrate communities [87, 9, 57]. The decreasing cost and increasing reliability have recently led to the application of camera traps for long-term, continuous deployment aiming at monitoring the entire wildlife communities across multiple seasons and years [124, 121, 2, 116]. Compared with conventional one-time or annual surveys, continuous monitoring reveals new insights into wildlife responses to local, regional, and global environmental changes and to conservation interventions. This scale of monitoring is particularly valuable for capturing responses to environmental perturbations as they occur [121, 102]. The 'Snapshot Serengeti' project (www.snapshotserengeti.org), which has operated continuously since 2010, is a flagship example of a long-term camera trap monitoring program. Over the last decade, this survey has gathered unprecedented longitudinal data that have significantly enhanced our understanding of the seasonal and inter-annual dynamics of the Serengeti ecosystem [124, 4, 88]. Projects of this magnitude have recently become increasingly common across eastern and southern Africa [116] and around the world [121].

Like short-term camera trap projects, the greatest logistical barrier to long-term monitoring with camera traps is also the overwhelming amount of human labor needed to annotate thousands or millions of wildlife images for ecological analysis [101, 3, 124, 86]. This annotation bottleneck creates a considerable mismatch between the pace of data collection and data processing, significantly curtailing the usefulness of camera trap data for ongoing conservation and monitoring efforts [3]. For example, a relatively modest camera trap survey (∼80 camera traps; [121]) captures millions of images a year. We estimate that it would take a single trained expert around 200 full-time workdays to annotate one million images. As such, hundreds of human annotators (e.g., experts, trained volunteers, and citizen scientists) are required to keep pace with image accumulation. This need is likely to grow exponentially over the coming decades as more monitoring sites are set up. While only one or two experts are needed to validate each wildlife image, it is common practice that multiple (5-20) volunteers or citizen scientists look at each image in order to produce a high-accuracy "consensus" classification (∼97% accurate compared to expert IDs; [124]). This duplication of effort needed to generate accurate results using volunteers further perpetuates the classification bottleneck.

## Automatic image recognition systems

As discussed in Chapter 1 and Chapter 2, the application of deep learning has the potential to improve the efficiency of processing wildlife imagery significantly, as a trained deep learning model can classify millions of images in a single day on a desktop computer [86, 127, 145]. However, before it becomes feasible to rely on deep learning to handle the mass of image data from large-scale long-term camera trap projects, several impediments must be overcome. In Chapter 2, we have discussed three challenges in avian recognition. While super-class challenge is relatively rare in camera trap datasets, the other two challenges also exist in long-term camera trap projects: 1) temporal changes in species composition at study sites due to migration, invasion, re-introduction, and extinction and 2) the long-tailed distribution of records across species (i.e., extreme imbalance in the number of images of different species). As discussed below, these issues also limit the ability of current AI models to accurately recognize species in camera trap datasets that are of significant interest to conservation practitioners.

### Changing species composition

The first challenge for long-term surveys is that new species may be detected on cameras in subsequent seasons or years, either because the species are rare and undetected in the previous survey periods [58] or because they are new to the system. Additionally, the species composition of ecological systems naturally varies through time through the process of succession [96]. Novel species are often of particular conservation concern, as they may represent recolonizing populations [81], reintroduced animals [130, 101], or harmful invasive species [20, 14].

In conventional deep learning, researchers focus on the performance of existing testing data while ignoring the potential for future changes in data composition [5]. In other words, deep learning models typically require datasets to be fixed in the number of categories (i.e., static), while in reality, long-term camera trap datasets are not constrained to certain numbers of species (i.e., dynamic).

Fine-tuning models through transfer learning is currently the best solution when new species populate a study area [148]. However, this process requires full annotation of newly-collected datasets, requiring considerable new human effort. This defeats the purpose of deep learning to reduce manual labor for long-term camera trap monitoring.

### Data from wildlife communities are long-tailed

As discussed in Chapter 2, wildlife communities typically contain many individuals of several common species and few individuals of many rare species, resulting in camera trap data with a long-tailed distribution. For example, in the dataset used in this chapter from Gorongosa National Park, Mozambique, $\sim$50K images ($>$ 60% of the animal images) are of `Baboon`, `Warthog`, and `Waterbuck`, while only 22 images are of `Pangolin` (a rare and protected species). This imbalance creates performance inconsistencies because deep learning success is derived from balanced training datasets (e.g., ImageNet [27]). For the Gorongosa dataset, a traditional deep learning approach (i.e., standard model) resulted in only 60% accuracy for a category with only 41 images (`Serval`) versus 88.8% performance for a species with 17,938 images (`Waterbuck`). This is a major issue because animals of particular conservation concern are typically rare [93], producing fewer images and therefore worse classification accuracy than common species. If such species are always misclassified, AI's practical benefits are limited.

## An iteratively updating recognition system

To overcome these two major issues of 1) changing species community composition and 2) long-tailed species distributions, we designed a deep learning recognition framework that is updated iteratively using limited human intervention. Human annotation is needed whenever images of species novel to the AI model appear in the data. Our goal, therefore, is to minimize the need for human intervention as much as possible by applying human annotation solely on difficult images or novel species while maximizing the recognition performance/accuracy of each model update procedure (i.e., update efficiency).

Traditionally, a deep learning model is applied to new batches of unannotated data collected during each time period to predict species classes. In our approach, we actively flag images that our model predicts with low-confidence as novel or unknown species. These low-confidence predictions are then selected for human annotation, while high-confidence predictions are accepted as accurate and used as pseudo-labels for future model updates. Then, the model is updated (i.e., retrained) based on both human annotations and pseudo-labels. To accommodate changing species communities, this procedure of active annotation and model update repeats each time new data are added to the collection (Figure 3.1). In terms of long-tailed distribution, we use the Open Long-tailed Recognition (OLTR)[74] method in our approach to balance the learning between abundant and scarce species. This component can reduce the number of predictions with low-confidence from scarce species.

As a case study, we trained a model on a camera trap dataset collected from Gorongosa National Park, Mozambique (an extension to the dataset used in Chapter 1) using this new method and produced significantly improved model update efficiency over traditional transfer learning approaches. Specifically, more than 80% of human effort was saved on annotating new data without sacrificing classification performance using our approach.

The dynamic nature of our algorithm maximizes learning and recognition efficiency by taking the best from both humans and machines within a synergistic collaboration. To the best of our knowledge, our model is the first framework that can be practically deployed for long-term camera trap monitoring studies.

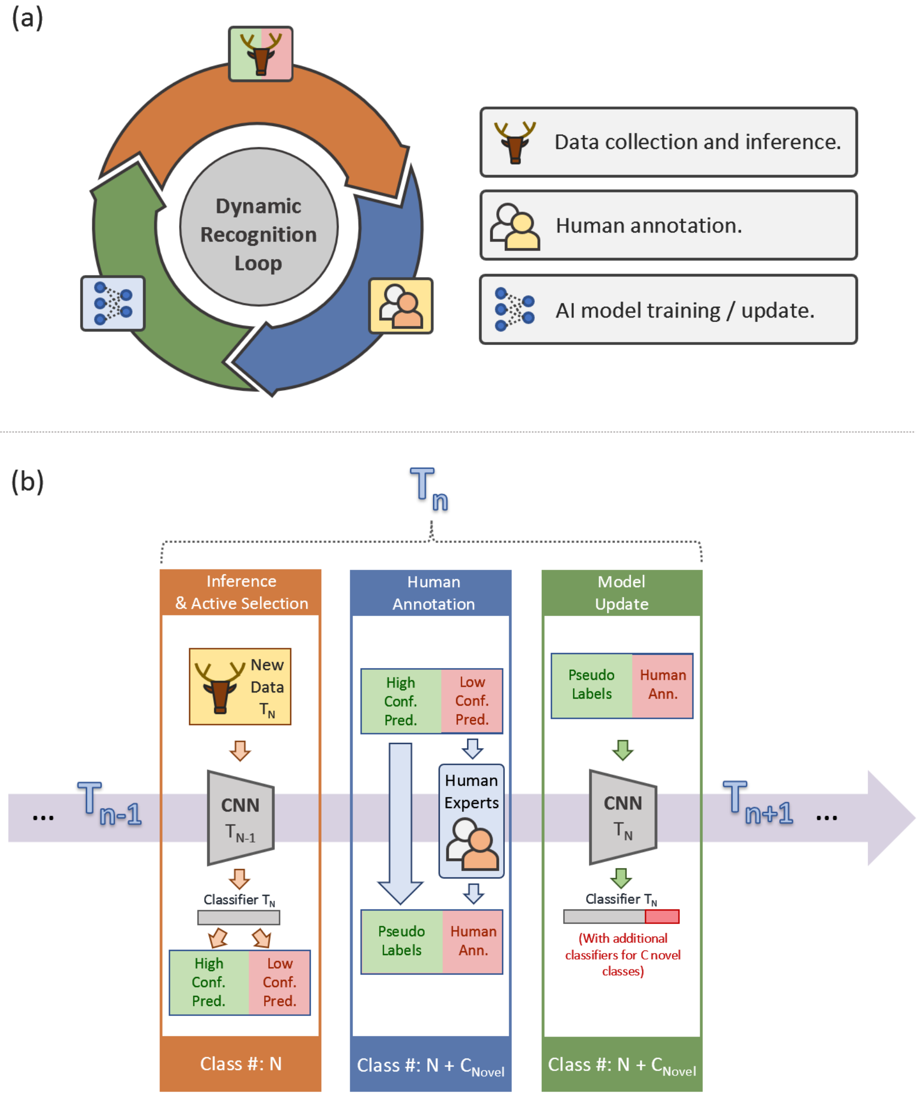


Figure 3.1: **a) Dynamic recognition loop.** In real-world applications, machine learning models do not stop at one training stage. As data collection progresses over time, there is a continuous cycle of inference, annotation, and model update. Every time a tranche of new data is added, pretrained models are applied to classify these data. When there are novel and difficult samples, human annotation is required, and the model needs to be updated to reflect the newly added data. **b) The progression of a realistic animal classification system.** Even if the trained model has high accuracy for the previous validation sets, there may be a difference in the classes between previous validation sets and current inference data (e.g., there may be novel categories in the newly collected data that do not exist in previous training and validation sets). Models, therefore, need to be updated over time. Here, we present a more practical procedure that can both maximize the utility of modern image recognition methods and minimize the dependence on manual annotations for model update. This procedure incorporates an active learning technique that actively selects low-confidence predictions for further human annotation while highly-confident predictions are kept as pseudo labels. Models are then updated according to both human annotations and pseudo labels. *Symbols: $T$ is time step. $CNN$ is convolutional neural network. $N$ is the total number of classes at time step $T_{n-1}$. $C_{Novel}$ is the number of novel classes at time step $T_n$.

# Data

## Data Categories

The dataset used in this chapter is an extension of the data used in Chapter 1, which only had 30 categories. For this chapter, we manually identified a total of 55 categories (i.e., species) in our data, including non-animal categories, such as `Ghost` (i.e., misfired images lacking animals), `Setup` (i.e., images with humans setting up the cameras), and `Fire`. There are 630,544 images in total. The complete list of these categories is in Figure 3.2, along with the number of images associated with each category. Some "vague" categories that human annotators were unable to accurately label because of the varying quality of camera trap images are also present, such as `Unknown Antelope` and `Unknown Bird`.

## Two groups of training and validation sets

To ensure sufficient training and validation data, we initially identified 41 of the most abundant categories in our camera trap dataset. The remaining 14 of the 55 categories were all tagged as `Unknown` and used to improve and validate the model's sensitivity to novel and difficult samples. Unlike the approach we did in Chapter 2, in this chapter we did not use these `Unknown` samples as an independent class during training. We randomly split the 41 categories (by trigger events) into **two groups of training and validation sets** (26 categories in the first group of data and 41 in the second group) to mimic periodical data collection from two sequential time periods (Figure 3.3). Detailed training and validation split information can be found in Appendix C.

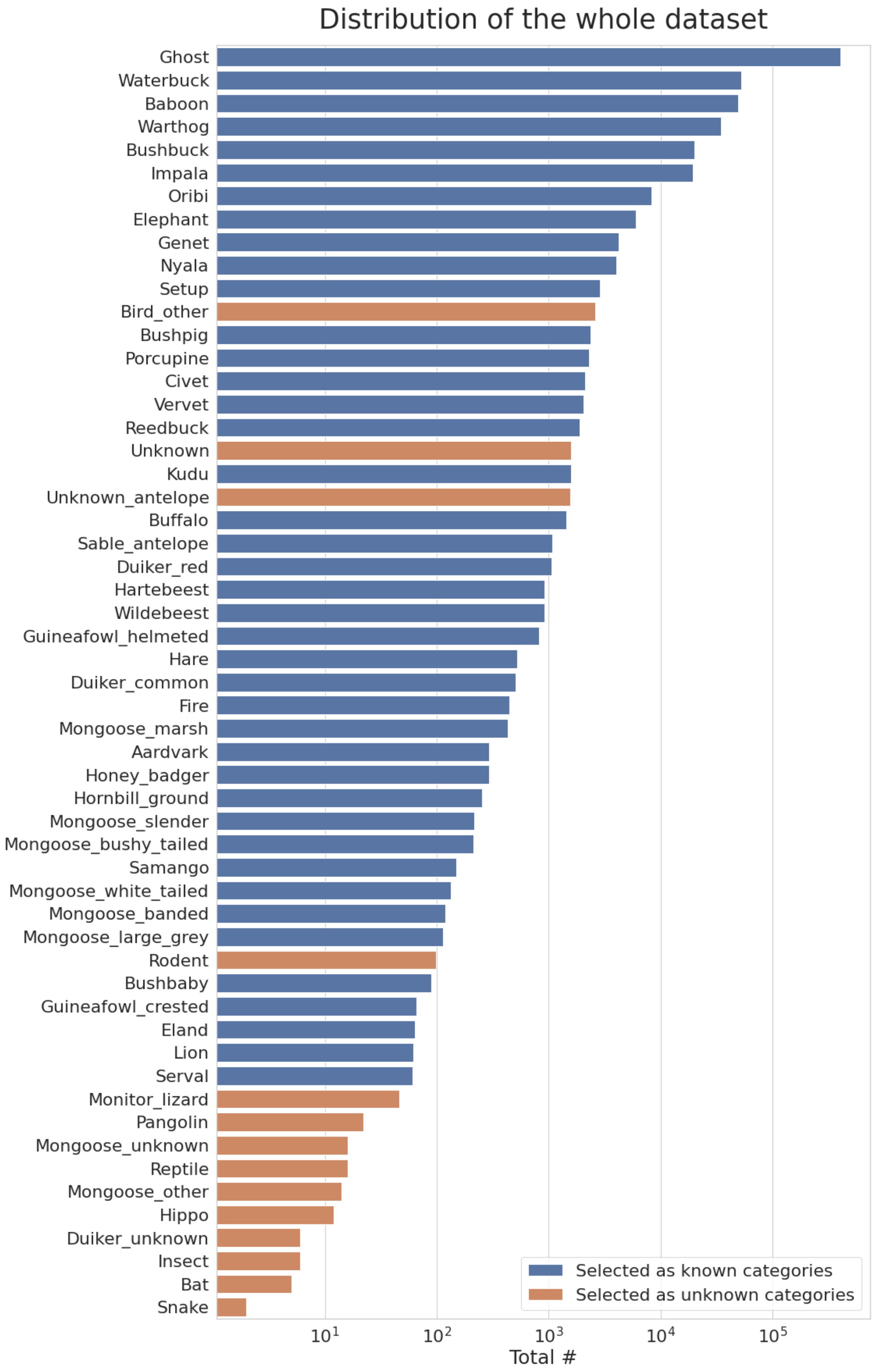


Figure 3.2: **The distribution of images across species in the entire camera trap dataset.** There are 55 categories in total. 14 categories were tagged as `Unknown` (colored in orange) and used to improve and validate our model's sensitivity to novel and difficult samples.

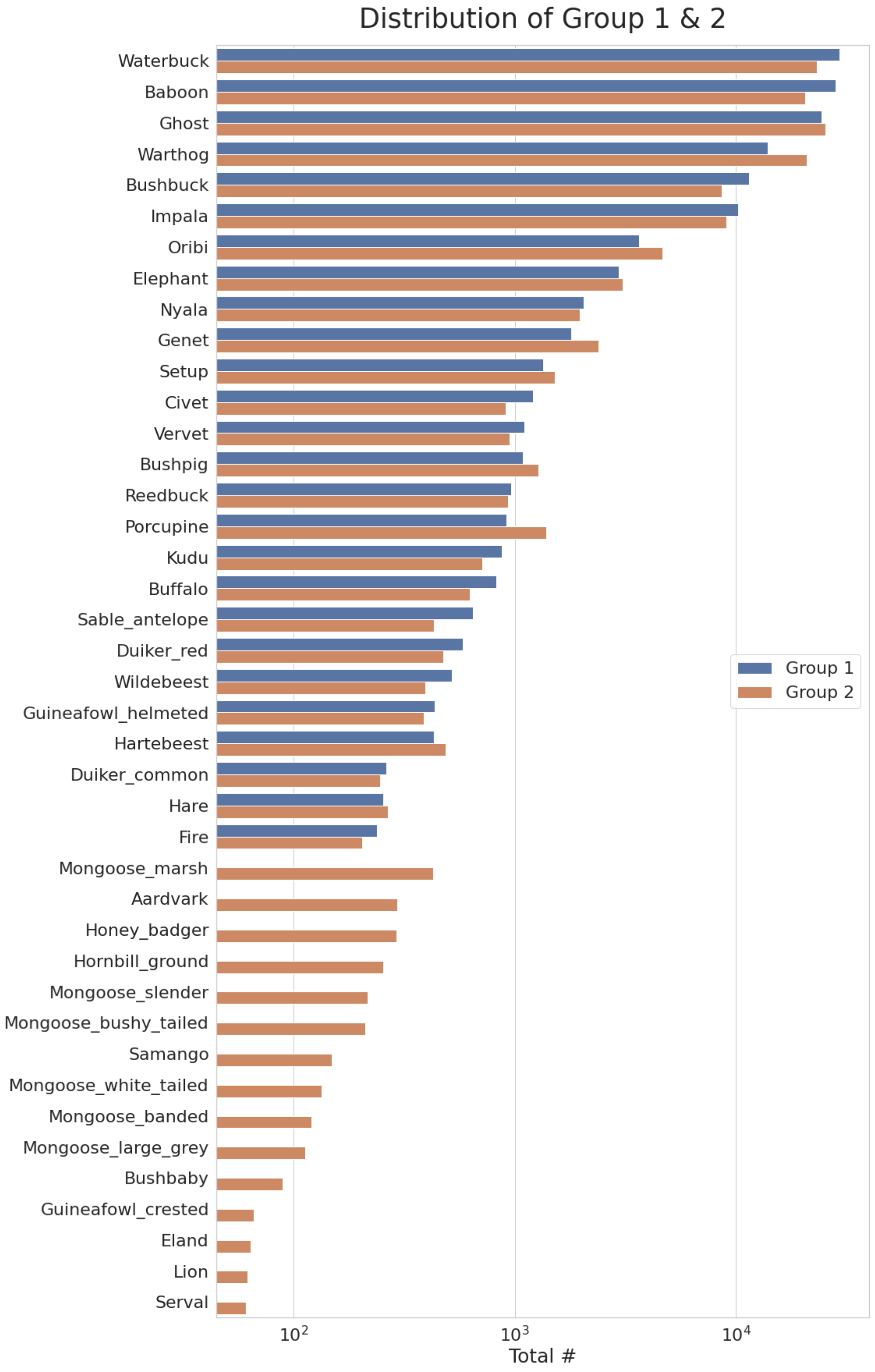


Figure 3.3: **The distribution of species across the two groups of data.** We split the dataset into two groups to mimic two sequential data collection seasons. In the first group, there are 26 categories (colored in blue). The second group has 41 categories. Group 1 was used in the first period experiment to train a baseline model, and Group 2 was used in the second period experiment to test and update the model.

# Methods

## Algorithm overview

Our approach has two major components: 1) active selection with human-in-the-loop and 2) model update using active data annotations. Specifically, at each time period when new data are collected, categories of images are predicted by deep learning models trained from previous periods with corresponding confidence levels. The model actively picks out low-confidence predictions for human annotation, while we accept high-confidence predictions as accurate. These predictions are used as pseudo-labels for further model updates or ecological analyses. Next, the model is updated (retrained) using both pseudo-labels and the newly acquired human annotations.

In our experiment, after updating the model, we evaluated model update efficiency and sensitivity to novel categories on a validation set. Specifically, we examined:

1. The overall validation accuracy of each category after the update (i.e., update performance).

2. The percentage of high-confidence predictions on the validation set (i.e., saved human effort for annotation).

3. The accuracy of high-confidence predictions.

4. The percentage of novel categories that were detected as low-confidence predictions (i.e., sensitivity to novelty).

The optimization of the algorithm aims to minimize human efforts (i.e., to maximize high-confidence percentage) and to maximize model update performance and high-confidence accuracy.

## Detailed pipeline for experiments

For experimental purposes, we divided our identification pipeline into two steps representing two time periods of data collection and the two groups of data curated in this chapter. The evaluation was focused on the second period when model update occurred. The overall experimental pipeline is illustrated in Figure 3.4. There are three major technical components in the framework: 1) energy-based loss [71] that improves the sensitivity to possible novel and difficult samples for active selection, 2) a pseudo-label-based semi-supervised procedure [66] for efficient model update from limited human annotations, and 3) open long-tailed recognition (OLTR) [74] that balances the learning of long-tailed distribution.

### Period 1

In the first period, we pre-trained an off-the-shelf/standard model (ResNet-50 model [46]) using the first group of data. After training, we adopted the energy-based loss [71] and data from the 14 “left-out” categories to fine-tune the classifier so it is more sensitive to novel and difficult samples.

### Period 2

In the second period, we first used the fine-tuned model from Period 1 to produce high- and low-confidence predictions from the group 2 training dataset, which were considered to be "newly collected." The confidence was calculated based on the *Helmholtz free energy* of each prediction [71]. Novel and difficult samples were distinguished using a preset energy threshold. Then, low-confidence predictions were annotated by humans while high-confidence predictions were accepted as pseudo-labels.

To update the model, we applied semi-supervised learning and OLTR, using both human annotations and pseudo-labels. Generally speaking, pseudo-label-based semi-supervised approaches iteratively update both classification models and pseudo-labels until the best performance on the validation sets is achieved [66]. The use of pseudo-labels also enables classification models to learn from the whole dataset instead of human-annotated data only. On the other hand, the OLTR approach balances the learning between abundant and scarce categories through an embedding space memory-based mechanism, where embedding memories of abundant categories are utilized to enhance the distinguishability of scarce categories that do not have enough samples to otherwise provide discriminative features [74]. See Appendix C for implementation details.

After the model was updated, the training samples from the 14 "left-out" categories were added to fine-tune the model's sensitivity to novel and difficult samples using energy-based loss as in Period 1.

### Future Periods

Because the framework is designed to aid long-term data collection and monitoring projects, the framework does not stop at Period 2. As time progresses, new data are collected. Users simply have to repeat the steps in Period 2 to pick out and annotate difficult/novel samples to update the model. In addition, since the framework is fully modular, when new techniques are developed, parts of the framework can be easily replaced for better performance. For example, if there are better methods for novel-category-detection, energy-based loss and confidence calculation can be replaced with no effect on the conceptual framework.

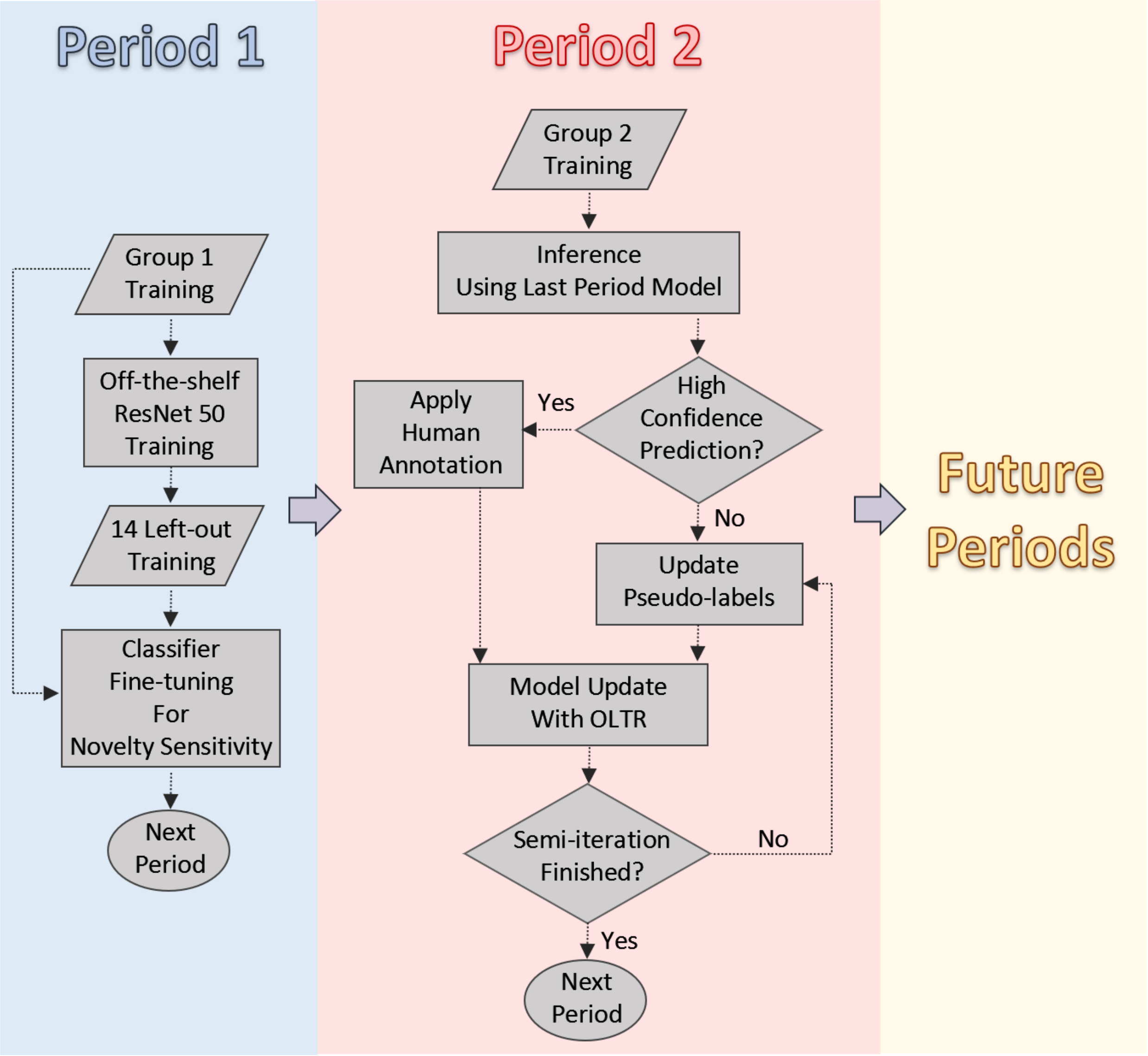


Figure 3.4: **The overall experimental pipeline of our framework.** In the first time step/period, a baseline model was trained using the group 1 training data with only 26 categories. Next, the classifier was fine-tuned using the 14 unknown categories and energy-based loss to increase the sensitivity to out-of-distribution categories. After the classifier was fine-tuned, the classifier was then used to classify the group 2 training data. Here, high-confidence predictions were trusted, while low-confidence predictions were flagged for human annotation. In the final step, both machine and human annotations were used to update the previous model with OLTR and semi-supervised techniques. Once the model was updated, the classifier was fine-tuned using energy-based loss again for out-of-distribution sensitivity.

# Results

## Period 1

In the first period, the model achieved an 81.2% average class accuracy on the validation set of group 1. 79.5% of the predictions were high-confidence, and of these predictions, the accuracy was 91.1% (Table 3.1 and Table 3.2). In terms of novel categories, in the validation phase, the model successfully detected 90.1% of the novel samples belonging to the 14 categories that were "left-out" of the training phase. In other words, 90.1% of the novel samples were predicted with low-confidence. In contrast, direct Softmax confidence (the most conventional way of calculating prediction confidence [48]) achieved a similar high-confidence accuracy as our model (91.5%) but only detected 59.3% novel samples.

Table 3.1: **Classification performance comparisons on validation sets of periods 1&2.**

| **Periods** | **Methods** | **Class Avg. Acc. (%)** | **Class Avg. Acc. On New Classes. (%)** |
|---|---|---|---|
| **1** | **Standard Model** | 81.2 | - |
| **2** | **Traditional transfer learning w/ full human ann.** | 75.8 | 63.9 |
| | **Our framework w/out semi-supervision and OLTR** | 69.2 | 61.2 |
| | **Our framework (Semi-OLTR)** | 77.2 | 68.1 |

Red color means higher performance on the **same** inference set.
w/ : with.
ann : annotation.
Avg. : Average.
Acc. : Accuracy.

Table 3.2: **Active selection performances of Period 1&2 with and without energy based function.**

| **Periods** | **Inference sets** | **Confidence Metrics** | **High Conf. Ratio (%)** | **High Conf. Acc. (%)** | **Novel Detect Ratio (%)** |
|---|---|---|---|---|---|
| **1** | **Group 1 Val.** | **Softmax** | 80.9 | 91.5 | 59.3 |
| | **Group 1 Val.** | **Energy (Ours)** | 79.5 | 91.1 | 90.1 |
| **2** | **Group 2 Train** | **Energy (Ours)** | 78.7 | 92.4 | 75.7 |
| | **Group 2 Val.** | **Softmax** | 71.2 | 90.1 | 70.5 |
| | **Group 2 Val.** | **Energy (Ours)** | 72.2 | 90.2 | 82.6 |

Red color means higher performance on the **same** inference set.
Conf. : Confidence.
Acc. : Accuracy.

### Period 2

In the group 2 training dataset, the model pretrained from Period 1 predicted 78.7% images with high-confidence, where the accuracy was 92.4%. 75.7% of the new categories in the group 2 training dataset were detected as low-confidence predictions (Table 3.2). As high-confidence predictions are trusted, 78.7% human effort was saved from annotating the group 2 training dataset because high-confidence predictions were accepted as accurate in our framework.

To update the model, group 2 training data predicted with low-confidence were checked by human experts and provided with manual annotations, and high-confidence samples were assigned model-predicted pseudo-labels. Overall, on the validation set of group 2, the model updated on both human annotations and pseudo-labels had an average class accuracy of 77.2% over the 41 categories. Compared to our method without human annotation (69.2%; second to the last row in Table 3.1), there was an 8% improvement. The model had 72.3% high-confidence predictions at a 90.2% accuracy in the validation set (see Table 3.1, Table 3.2). In addition, it had an 82.6% novel sample detection rate (i.e., flagged as low-confidence predictions) in the validation data of the 14 "left-out" categories (see the last column of Table 3.2).

### Comparison with traditional transfer learning

Our model was significantly more data efficient (i.e., fewer data required for similar performance) than traditional transfer learning methods in several respects. Compared to traditional transfer learning, which used full human annotations of the group 2 training dataset, our method only involved human annotation of 21.3% of the group 2 samples. Even with less human annotation, our method still achieved better overall class average accuracy (77.2% vs. 75.8% for traditional transfer learning; Table 3.1, Table 3.2, and Figure 3.5). Our model also performed better than direct transfer learning for classifying the 15 new categories from Group 2 (with an average of 4.2% accuracy improvement; Table 3.1 and 3.2).

### Practical deployment

Our new framework showcases the powerful potential of deep learning for long-term ecological applications while employing a novel practical approach that greatly reduces the manual annotation burden. To validate the practical benefits, we deployed the model to classify a new set of data gathered from the same camera trap monitoring sites (Gorongosa National Park, Mozambique) after the group 1 and 2 datasets were collected (see Appendix C for details). The new dataset was unannotated, unanalyzed, and contains 623,333 images in total. Images were predicted with the same active selection procedure, and 78.7% of the predictions were considered high-confidence. Thus only 21.3% of these newly-collected data required human annotation (or 78.7% of the human effort, and ultimately annotation cost, was saved).

To validate the robustness of model performance, two experts (KMG and MSP) confirmed the accuracy of 1000 randomly-selected high-confidence predictions (i.e., those that were accepted as accurate). As a result, our model predictions were 88.6% accurate with respect to expert classifications. Statistically, ∼88% automatic accuracy is already sufficient to

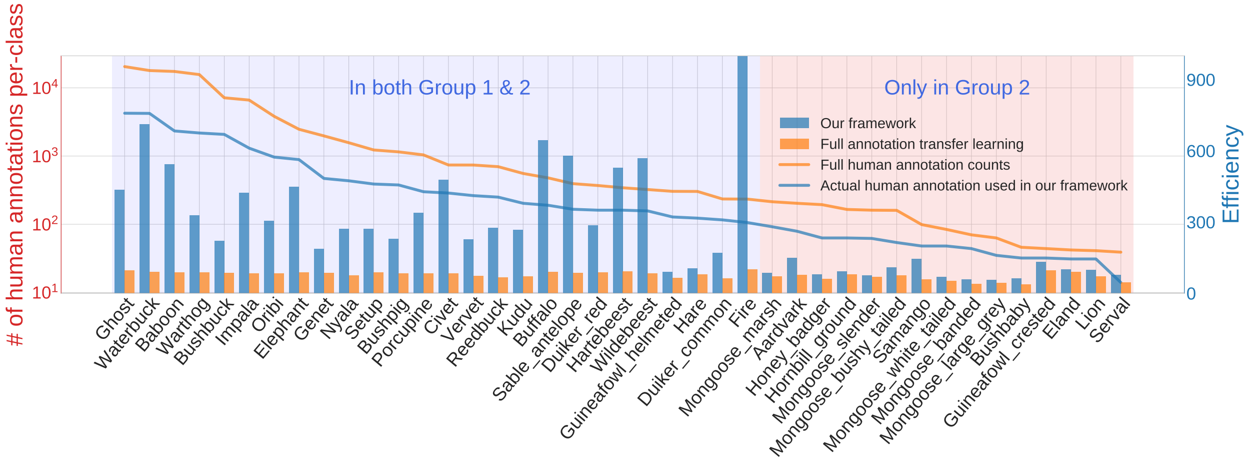


Figure 3.5: **Label efficiency comparison with transfer learning on Group 2 validation set (ordered with respect to training sample size).** To examine label efficiency (a measure of accuracy given the number of annotations) after we updated our model in Period 2, we calculated validation accuracy over the percentage of used training annotations of each category. In other words, we define label efficiency: $\text{Efficiency}_i = \text{Validation Accuracy}_i/(\#\text{ of training annotation}_i/\#\text{ of full annotation}_i)$, where $i$ is the category index. The higher the value is, the more efficient the model is at learning corresponding categories, and the less training data are needed to achieve comparable if not better performance of full manual annotations. In the figure, we illustrate label efficiencies of all categories that exist in the Group 2 training and validation set. The blue bars represent our model's label efficiencies of each category. The orange bars represent baseline efficiencies for comparison, where full annotations were used with the traditional transfer learning method (i.e., $\#\text{ of training annotation}_i/\#\text{ of full annotation}_i = 1$). The blue and orange lines are annotation counts of each category, where orange represents full annotations, and blue represents used human annotations in our Period 2 model update procedure. For categories that exist in both the Group 1 & 2 training sets (i.e., known categories; on the left, with a blue background), the efficiency was significantly higher than the baselines across all categories. For categories that only exist in Group 2 datasets (i.e., they were absent in the Group 1 training and validation set; novel categories; on the right, with orange background), because the model is designed to use as many annotations as possible the novelty of these categories, $\#\text{ of training annotation}_i/\#\text{ of full annotation}_i$ of these categories were close to 1. However, our model still had relatively higher efficiency than the full annotation transfer learning model across all the novel categories because our model had higher validation accuracy with a similar amount of training annotations.

help alleviate the data bottleneck encountered in typical camera trap monitoring projects compared to expert accuracy.

In terms of future model update, the model can be further updated and validated on new datasets using the same procedure as Period 2, where a new validation set can be created using a mix of previous validation sets (validation of groups 1 & 2) and the newly acquired human annotations. In addition, the same random verification by human experts on high-confidence predictions can also be applied to avoid performance corruptions (i.e., increased misclassifications in high-confidence predictions).

### Invasive and recolonizing species

One of the significant advances made by our framework is the ability to flag new or rare species that may have particular conservation importance. Our new dataset contains two novel species (`Leopard` and `African Wild Dog`) to test the model's sensitivity to novel categories. The former naturally re-colonized the study area while the latter was re-introduced as a part of ongoing conservation efforts. There are 24 and 5 images for `African Wild Dog` and `Leopard`, respectively. The model successfully detected 20 (83.3%) `African Wild Dog` images and 4 (80.0%) `Leopard` images, demonstrating its capacity to recognize important novel species in continuous monitoring periods.

## Discussion

### Misclassifications

Two types of misclassifications/failures occurred in our experiments: 1) low-confidence predictions that were not novel species, and 2) high-confidence predictions that differed from human-supplied annotations.

First, there are several ways in which our model was unable to accurately identify samples from known species with high-confidence (Figure 3.6a). A common reason for low-confidence predictions was the difficulty of distinguishing animals from the background. For example, Figure 3.6a.i depicts an antelope obscured by darkness at night, making it difficult for the model to classify confidently. However, rather than making misclassifications as would occur in traditional AI approaches [46], our model considered the prediction low-confidence and flagged the image for review. In our approach, most of these difficult samples are more likely to be flagged as low-confidence predictions for further human evaluation (annotation) rather than assigned random labels—a practice that can potentially bias further data analysis and inference.

In the second type of model failure, predictions of images predicted with high-confidence differed from the original annotations (Figure 3.6b). We note that these images were initially classified by trained volunteers who may not have correctly annotated all samples as accurately as wildlife experts. Surprisingly, most of the confident predictions are proven correct after human experts' re-evaluation (KMG and MSP). For example, Figure 3.6b.iv was originally labeled as `Warthog`, although there is no warthog present. However, there is a vervet monkey in the lower left of the frame that was missed by the human classifiers. The model not only detected the previously unobserved animal but also correctly identified the species.

Thus, these "failures" actually demonstrate the robustness and flexibility of our framework. As both human annotations and machine predictions can be wrong, a mutual interaction between humans and machines can benefit long-term performance of the recognition system. For example, picking out low-confidence samples like the ones in Figure 3.6b prevents producing low quality predictions that can cause bias in camera trap analyses. Further, applying validated human annotations on these samples can help improve the identification capacity of the model as it needs to recognize more difficult samples during model updates. On the other hand, when the model is highly confident, it can be more accurate than average human annotators, as evidenced by the examples given in Figure 3.6b.ii, iv, and v. In other words, some of the human mistakes are prevented, such that the annotation quality for future model update and camera trap analyses are improved. On the other hand, as we acknowledge that in some cases the model will make high-confidence misclassifications, we can apply random periodical verification by human experts on high-confidence predictions (similar to what we did in the Practical deployment section) to ensure that these errors do not propagate through repeated training.

### The need for human-in-the-loop

Our framework demonstrates the unique merit of combining machine intelligence and human intelligence. As Figure 3.6c illustrates, machine intelligence, when trained on large datasets to distill visual associations and class similarities, can quickly match visual patterns with high confidence [27]. Human intelligence, on the other hand, excels at being able to recognize fragmented samples based on prior experience, context clues, and additional knowledge. Increasingly, we are moving towards applying computer vision systems to real-world scenarios, with unknown classes [74], unknown domains [75], and constantly-updating environments. It is, therefore, crucial to develop effective algorithms that can handle dynamic data streams. Human-in-the-loop provides a natural and effective way to integrate the two types of perceptual ability (i.e., human & machine), resulting in a synergism that improves the efficiency of the overall recognition system.

### Extensions and future directions

Our framework is fully modular and can be easily upgraded with more sophisticated model designs. For example, models with deeper networks can be employed for better classification generalization, more sophisticated semi-supervised training protocols can be adopted for better learning from pseudo-labels, and better novelty detection techniques can be used for better active selection.

Future directions include extending our framework to handle multi-label and multi-domain scenarios. The current approach was developed for single-label recognition (i.e., each image only represents one single species). However, it would be desirable to recognize multiple species within the same view in real-world camera trap setups. Further, our framework is expected to be deployed in diverse locations with different landscapes. Therefore, our methodology can be more scalable with the ability to handle multiple environmental domains than existing methodologies. In addition, our method will be incorporated in a user-friendly interface, such that users without knowledge of Python can use it.

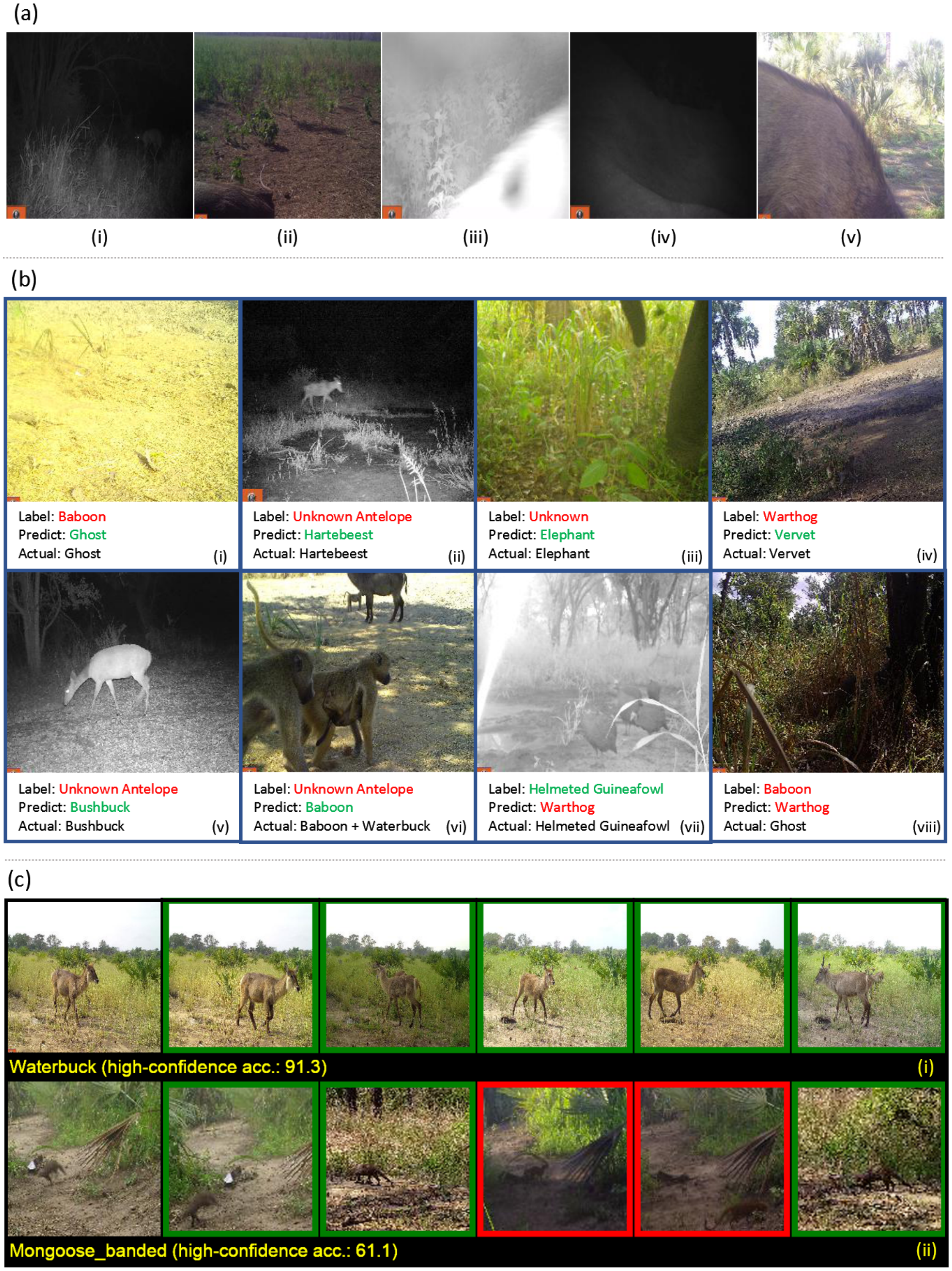

(a)
(i)
(ii)
(iii)
(iv)
(v)
(b)
Label: Baboon
Predict: Ghost
Actual: Ghost
(i)
Label: Unknown Antelope
Predict: Hartebeest
Actual: Hartebeest
(ii)
Label: Unknown
Predict: Elephant
Actual: Elephant
(iii)
Label: Warthog
Predict: Vervet
Actual: Vervet
(iv)
Label: Unknown Antelope
Predict: Bushbuck
Actual: Bushbuck
(v)
Label: Unknown Antelope
Predict: Baboon
Actual: Baboon + Waterbuck
(vi)
Label: Helmeted Guineafowl
Predict: Warthog
Actual: Helmeted Guineafowl
(vii)
Label: Baboon
Predict: Warthog
Actual: Ghost
(viii)
(c)
Waterbuck (high-confidence acc.: 91.3)
(i)
Mongoose_banded (high-confidence acc.: 61.1)
(ii)

Figure 3.6: **(a) Examples of low-confidence predictions.** In most of the cases, the model had low-confidence predictions on images with distorted, partially visible (panel ii∼v) or obscured animals (panel i). It can be incredibly difficult, if not impossible, for both humans or machines to accurately identify the animal species. **(b) Examples of high-confidence predictions that did not match the original annotations.** Many high-confidence predictions that were flagged as incorrect based on validation labels (provided by students and citizen scientists) were correct upon closer inspection by wildlife experts (KMG and MSP). For example, in Panel (i), an empty image originally mislabeled as `Baboon`, was correctly classified by our method as empty. In panel (ii), although the animal is distant from the camera in a dark environment, the model successfully identified `Hartebeest`, while the human-supplied label was `Unknown Antelope`. In panel (iii), the model successfully identified the elephant only based on the trunk and leg, while human volunteers originally classified the image as `Unknown`. In panel (iv), a vervet monkey was correctly detected and classified in an image originally (incorrectly) labeled as `Warthog` by human annotators. Panel (v) was originally labeled as `Unknown` by human annotators, but based on the body shape and white markings on the rear, the model correctly recognized the animal as `Bushbuck`. Panel (vi) is an example of multiple species in the same scene. Although the model did not have the capacity to deal with multi-species samples, as `Baboon` is obviously the major component of this image, the prediction is reasonable. On the other hand, these examples above do not mean that the model always makes correct predictions when highly confident. Panels (vii) and (viii) are two typical examples where the model made mistakes due to the obscured nature of these images. Red text indicates "wrong" classification, and green text indicates correct classification. **(c) Two examples of image retrieval based on feature space similarity.** Machine intelligence largely depends on visual similarity associations learned from large-scale datasets to classify animal species. These two examples illustrate image retrieval based on the Euclidean distances of the feature vectors (i.e., outputs of the global average pooling layer of the ResNet model used in this Chapter, which is of dimension 2,048 in Euclidean space). For each anchor image (the leftmost image of each row), we show five closest (i.e., most similar) samples in terms of Euclidean distance within the validation set of Group 2. Green color means correct predictions, and red color means "wrong" predictions (based on the original annotations). For example, in sequence (i), samples with similar visual appearances are usually from the same species (`Waterbuck`). However, in sequence (ii), the two most similar images (according to our model) to the `Banded Mongoose` anchor image are actually not `Banded Mongoose` but `Slender Mongoose`. The model misclassified these two samples based on their similarities to the other `Banded Mongoose` images.

# Appendix A: Supplementary material for Chapter 1

## Data

### Data background

The camera trap data come from a long-term research program in Gorongosa National Park, Mozambique (18.8154°S, 34.4963°E). The data set used in Chapter 1 was collected from June to November of 2016. The goal of this program is to examine the spatial distribution of large mammal species in the park and to monitor the restoration of the park's wildlife following decades of civil war. The 3,700 km$^2$ park encompasses a range of habitats, including a mix of grassland, open woodland, and closed forest. KMG placed 60 motion-activated Bushnell TrophyCam and Essential E2 cameras in a 300 km$^2$ area in the southern area of the park. Each camera was mounted on a tree within 100 meters of the center of a 5 km$^2$ hexagonal grid cell, facing an animal trail or open area with signs of animal activity. To minimize false triggers, cameras were set in shaded, south-facing sites that were clear of tall grass. Cameras were set to take 2 photographs per detection with an interval of 30 seconds between photograph bursts.

### Trained researcher classification and visual descriptors

The species in all images were manually classified and annotated independently by 2 different researchers trained on a list of example images and corresponding visual descriptors of each species; this list was created by KMG before the manual annotation and was iteratively updated as the annotation progressed. All classifications were confirmed by KMG prior to this project.

We conducted a survey to determine the features regularly used by humans to identify each of the 20 species in this study. For each species, respondents were asked to select features that they regularly look for and/or use as clear diagnostic features that identify the species and could select as many descriptors as they wanted. We provided respondents with KMG's list of all visual descriptors used in training materials and included an option of adding additional descriptors not mentioned. The survey had 13 respondents who have extensive experience classifying camera trap images from Gorongosa National Park, including those used in this study. KMG selected the participants, and they included 5 undergraduate research apprentices, three other researchers affiliated with Gorongosa National Park, and three trained citizen scientists. KMG and ZM also completed the survey. We considered a feature to be regularly used by humans if at least 5 of the 13 respondents selected it (Table A.3).

### Data preprocessing

We first grouped the images by camera shooting events. In our camera trap program, when the motion sensors detect motion at each shooting event, the cameras capture 2 sequential images within 1 second. Therefore, image pairs of the same shooting events often are similar in appearance, and the training performance of the model can be overestimated if images from the same image pair are separated into training and test sets. Thus, we maintained the image pairs in the analysis to prevent a negative bias of the CNN learning process. We then randomly split the image groups into training, validation, and testing sets with 85%, 5%, and 10% of the data sets.

## Methods and implementation details

### Model implementation

We trained a VGG-16 [115] CNN architecture to classify camera trap images with class-aware sampling [113]. The output of the CNN classifier is a 20-dimension vector, with each dimension representing the classification probability for an animal species (classification score). The use of class-aware sampling helps to improve the classification accuracy for imbalanced data sets.

We made use of PyTorch [89], a deep learning framework, to implement and train the CNN. The weights were initialized from an ImageNet [26] pretrained model. The initial learning rate was 0.01, which decreased every 15 epochs. The best model was obtained at epoch 40, where the classification accuracy on the validation data set was the highest. The loss function used to train the CNN was Softmax cross-entropy loss. All the input images for training were first downsized to 256x256, then were randomly cropped to 224x224 with a random horizontal flip at a rate of 0.5. Values of the hyperparameters used for training are listed in Table A.1.

### Guided Grad-CAM (GG-CAM)

GG-CAM is a method that combines the outputs of Grad-CAM and GBP [111]:

Grad-CAM generates coarse, discriminative regions according to animal species. It is calculated as the rectified linear units (i.e., $\max\{0, x\}$) of the weighted sum of the response maps from the last convolutional layer (Eq. 3.1). The weighted sum is based on the importance value $\alpha_k$ (importance value of the $k_{th}$ neuron) of each neuron (neuron importance) in that layer of the response map, $A^k$, where its $ij^{th}$ element is $A^k_{ij}$, for a total number of element $Z$. If $y$ is the prediction score of animal A before the Softmax layer, then Grad-CAM

Table A.1: **Hyperparameters**

| Parameters | Values |
|---|---|
| Input image size: | 256x256 |
| Random crop size: | 224x224 |
| Random horizontal flip rate: | 0.5 |
| Batch size: | 256 |
| Training epoch: | 40 |
| Initial learning rate: | 0.01 |
| Momentum: | 0.9 |
| Learning rate reduce at: | every 15 epochs |
| Learning rate reduce by: | 0.1 |
| Regularization: | None |

is computed using the following equations:

$$\text{Grad-CAM} = \max\Big\{0, \textstyle\sum_k \alpha_k A^k\Big\} \tag{3.1}$$

$$\alpha_k = \frac{1}{Z}\textstyle\sum_i \sum_j \dfrac{\partial y}{\partial A^k_{ij}} \tag{3.2}$$

GBP is a method that captures non-class-discriminative details of visual components that are important to the network overall. It is calculated as the gradient of the output response maps of the last convolutional layer with respect to the input images, with only positive gradients and positive response elements (Eq. 3.3). Suppose $R^l$ is the GBP product of the $l^{th}$ layer, then it is calculated in terms of the response maps $f^l$ of the $l^{th}$ layer, and the response maps $f^{\text{out}}$ of the last convolutional layer. Specifically, defining $f^{l'} = \max\{0, f^l\}$, the equation is:

$$R^l = f^{l'} \times \max\left\{0, \frac{\partial f^{\text{out}}}{\partial f^{l'}}\right\} \times \frac{\partial f^{\text{out}}}{\partial f^{l'}} \tag{3.3}$$

After training the model, we can fix the model weights and use Eq. 3.1 to calculate the Grad-CAM of the input. We can also use (Eq. 3.3) to generate the GBP results of the input with the trained model. Once Grad-CAM and GBP are generated, we can calculate the Hadamard product (a.k.a. element-wise multiplication) of Grad-CAM and GBP. GG-CAM is the normalized output of the Hadmard product (Figure 1.4).

## Similarity between extracted features and human visual descriptors

We also calculated the Dice similarity coefficient [118] of the extracted features and the corresponding visual descriptors to get a quantitative sense of how similar the model was to humans when making classifications. To calculate the similarity between extracted features and human descriptors, we first did the feature matching, which was agreed upon by up to 4 authors (ZM, KMG, ZL, and MSN). Then we used the Dice similarity coefficient (DSC $\in [0, 1]$) to calculate the similarity between the 2 sets of features. Suppose $M_C$ and $H_C$ are machine-extracted feature set and human descriptor set of animal $C$, the DSC is calculated as:

$$\text{DSC} = \frac{2 \cdot |M_C \cap H_C|}{|M_C| + |H_C|} \tag{3.4}$$

$|\cdot|$ is the cardinality of the sets.

## Mutual information

Next, we demonstrate our approach to inspecting within-species animal discriminative features based on common neuron importance. Each neuron in the network has a response to certain parts of the input images. Classification of images is based on a combination of neuron responses. In addition, certain neurons are more important than others for classification per species. We assume that the responses from these neurons can be regarded as common within-species features.

We use Mutual Information (MI) [78], a method commonly used to find information shared between variables [8, 91], on the neuron importance (Eq. 3.2) (normalized from 0 to 1) from the last convolutional layer across the data (Eq. 3.5). We calculated $I(U < C)$, the MI for neuron $U$ and animal species $C$, as follows. Suppose $N_{11}$ and $N_{01}$ are the number of images of $C$, where $U$ has neuron importance $> 0.5$ and $\leq 0.5$, respectively. Further suppose $N_{10}$ and $N_{00}$ are the number of images that are not C, where U has neuron importance $> 0.5$ and $\leq 0.5$, respectively. Defining $N_{1\cdot} = N_{10} + N_{11}$, $N_{\cdot 1} = N_{11} + N_{01}$, $N_{0\cdot} = N_{00} + N_{01}$, $N_{\cdot 0} = N_{00} + N_{10}$, and $N = N_{00} + N_{01} + N_{10} + N_{11}$ it then follows that

$$\begin{aligned} I(U,C) = & \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_{1\cdot}N_{\cdot 1}} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_{0\cdot}N_{\cdot 1}} \\ & + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_{1\cdot}N_{\cdot 0}} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_{0\cdot}N_{\cdot 0}} \end{aligned} \tag{3.5}$$

We calculated the class-wise MI scores using 6000 randomly selected images (300 images per class). Each neuron has 20 different MI scores for each class. After the calculation, we selected 9 images of each class that had the highest responses to each neuron. The results are illustrated in Figure A.1 for image patches with the highest responses to the neurons with the top 1 to top 5 mutual information scores of each species.

## Interspecific visual similarities and species unfamiliarity

To inspect visual similarities between animal species, we generated a visual similarity tree of all species by implementing hierarchical clustering on the feature vectors before the classifier layer (i.e., the output of the last fully-connected layer before the classifier layer). Firstly, we extracted the feature vectors of 6000 randomly selected training images and applied Principal Component Analysis (PCA) to compress the 4096-dimension feature vectors to 128 dimensions for computational simplicity. We then computed the average interspecific Euclidean distances between every pair of the feature centroids of the 20 species. Finally, we processed the interspecific distances using a hierarchical clustering method with the Ward variance minimization algorithm [84]. The leaves in the dendrogram can be regarded as the feature vector centroids of the 20 classes (Figure 1.8).

To calculate the relative unfamiliarity of 30 animals species, we first incorporated images of the 10 excluded species into the testing data set and performed a 10-round random selection. In each round, we randomly selected 20 images. This was because we only had 28 images for the rarest species (Pangolin), and we wanted to keep the testing data balanced. We calculated the Euclidean distances of the feature vectors of these images to the 20 feature centroids that constructed the dendrogram. Then the relative unfamiliarity was calculated as the mean distance of these feature vectors to their closest centroids across the 10-round random selection.

# Additional results

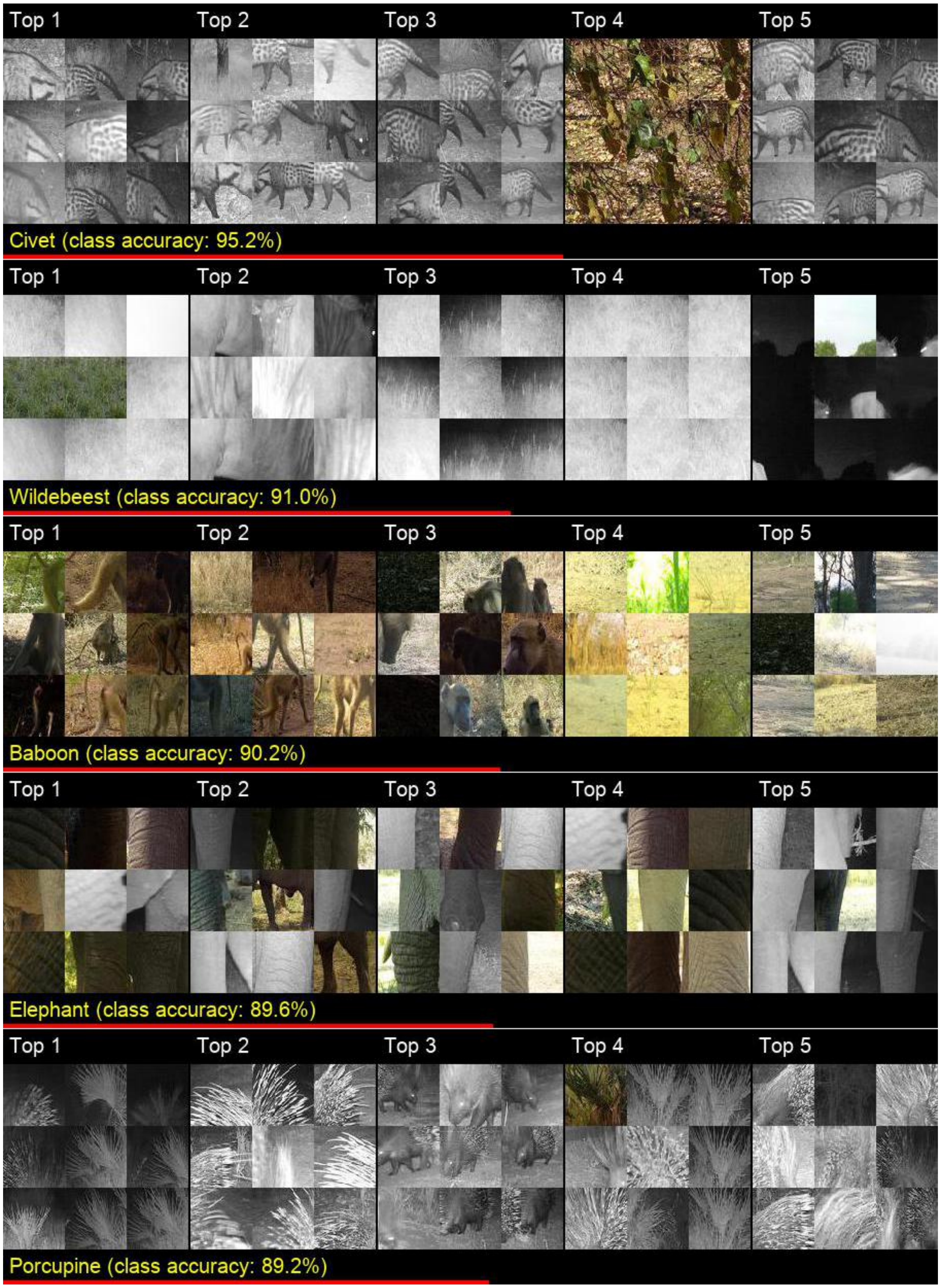
## Confusion matrix

The confusion matrix of the test results of our VGG-16 is reported in Table A.2. We can see that classes with a dominant amount of data (e.g., Baboon, Waterbuck, and Warthog) had a high recall but a low precision. This is because the model cannot generalize discriminative features from classes with a limited amount of training data, and thus data from these classes are more likely to be classified as dominant classes and cause higher false-positive rates. This pattern has been discussed in detail by imbalanced classification and long-tail distribution studies such as [135].

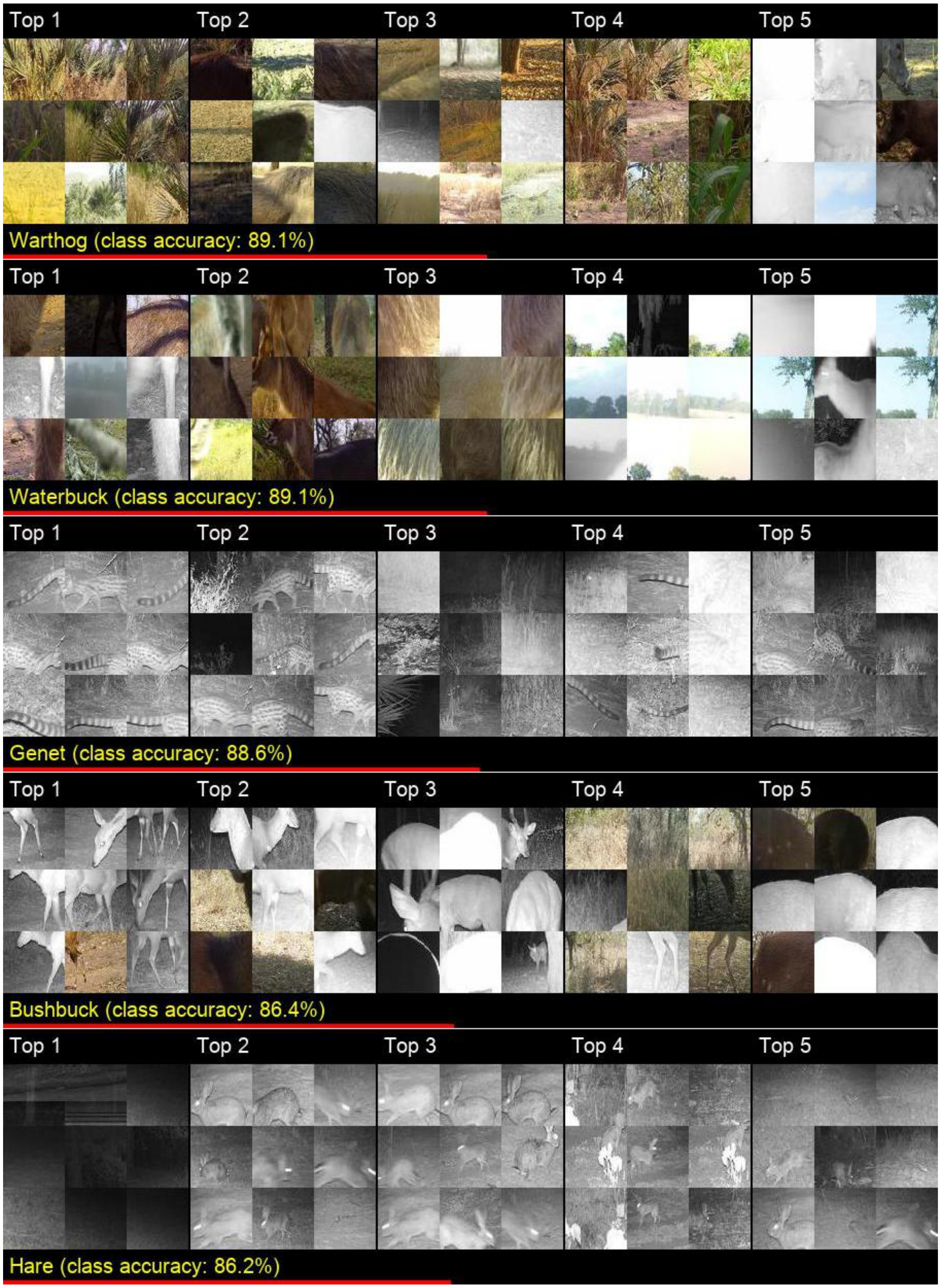
Table A.2: **Confusion matrix in percentage (%), where the rows represent actual classes and columns represent predicted classes [95]. The numeric column headings represent: 0: baboon, 1: buffalo; 2: bushbuck; 3: bushpig; 4: civet; 5: elephant; 6: genet; 7: hare; 8: hartebeest; 9: impala; 10: kudu; 11: nyala; 12: oribi; 13: porcupine; 14: reedbuck; 15: sable Antelope; 16: vervet Monkey; 17: warthog; 18: waterbuck; 19: wildebeest.**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 90.2 | 0.2 | 0.5 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 0.0 | 0.8 | 0.6 | 0.0 | 0.1 | 0.0 | 0.4 | 2.9 | 2.3 | 0.0 |
| 1 | 4.7 | 79.1 | 2.0 | 0.0 | 0.0 | 4.7 | 0.0 | 0.0 | 0.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 0.0 | 2.7 | 0.0 | 0.7 | 4.1 | 0.7 |
| 2 | 2.6 | 0.0 | 86.5 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.0 | 1.6 | 0.3 | 0.8 | 1.5 | 0.5 | 0.4 | 0.0 | 0.0 | 1.2 | 2.5 | 0.0 |
| 3 | 2.9 | 0.0 | 5.3 | 78.7 | 0.0 | 0.0 | 0.8 | 0.0 | 0.0 | 0.8 | 0.4 | 0.0 | 2.0 | 1.6 | 0.0 | 0.0 | 0.0 | 4.1 | 2.5 | 0.4 |
| 4 | 0.7 | 0.0 | 1.1 | 1.1 | 95.2 | 0.0 | 0.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 0.0 | 0.0 |
| 5 | 2.6 | 0.0 | 1.3 | 0.2 | 0.0 | 89.6 | 0.2 | 0.3 | 0.0 | 0.2 | 0.0 | 0.0 | 0.0 | 0.3 | 0.0 | 0.3 | 0.0 | 3.0 | 1.8 | 0.2 |
| 6 | 0.0 | 0.0 | 2.9 | 0.6 | 1.0 | 0.0 | 88.6 | 1.1 | 0.0 | 0.4 | 0.0 | 0.2 | 1.3 | 1.5 | 0.4 | 0.2 | 0.0 | 1.0 | 1.0 | 0.0 |
| 7 | 0.0 | 0.0 | 5.7 | 0.0 | 0.0 | 0.0 | 2.3 | 86.2 | 0.0 | 0.0 | 0.0 | 0.0 | 2.3 | 1.1 | 0.0 | 0.0 | 0.0 | 0.0 | 2.3 | 0.0 |
| 8 | 1.1 | 1.1 | 1.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 80.9 | 4.5 | 1.1 | 2.2 | 0.0 | 0.0 | 1.1 | 0.0 | 0.0 | 0.0 | 6.7 | 0.0 |
| 9 | 3.7 | 0.0 | 1.4 | 0.2 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 85.1 | 0.1 | 0.6 | 2.1 | 0.0 | 0.5 | 0.0 | 0.7 | 1.2 | 4.1 | 0.1 |
| 10 | 9.2 | 0.0 | 1.1 | 1.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.2 | 67.0 | 3.8 | 1.1 | 0.0 | 0.0 | 0.0 | 0.0 | 3.8 | 9.2 | 0.5 |
| 11 | 4.1 | 0.0 | 6.9 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 3.9 | 1.4 | 76.0 | 0.0 | 0.2 | 0.0 | 0.5 | 0.2 | 0.2 | 5.8 | 0.2 |
| 12 | 2.4 | 0.0 | 3.4 | 0.0 | 0.0 | 0.0 | 0.3 | 0.2 | 0.2 | 2.6 | 0.3 | 0.0 | 85.5 | 0.2 | 0.5 | 0.0 | 0.0 | 1.7 | 2.8 | 0.0 |
| 13 | 0.8 | 0.0 | 3.5 | 1.1 | 0.0 | 0.3 | 1.9 | 0.5 | 0.0 | 0.8 | 0.0 | 0.0 | 0.0 | 89.3 | 0.0 | 0.0 | 0.0 | 0.5 | 1.3 | 0.0 |
| 14 | 1.1 | 0.0 | 11.4 | 0.0 | 0.0 | 0.0 | 1.7 | 0.0 | 0.0 | 11.4 | 0.6 | 0.6 | 8.0 | 0.0 | 54.3 | 0.0 | 0.0 | 0.6 | 10.3 | 0.0 |
| 15 | 1.7 | 3.3 | 3.3 | 0.8 | 0.0 | 0.8 | 0.0 | 0.0 | 1.7 | 1.7 | 0.8 | 0.8 | 0.0 | 0.0 | 0.0 | 73.6 | 0.0 | 1.7 | 9.9 | 0.0 |
| 16 | 8.0 | 0.0 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 0.0 | 0.9 | 0.3 | 0.0 | 0.0 | 0.0 | 84.3 | 1.2 | 3.1 | 0.0 |
| 17 | 4.8 | 0.1 | 0.9 | 0.2 | 0.1 | 0.3 | 0.0 | 0.0 | 0.0 | 0.9 | 0.0 | 0.2 | 0.8 | 0.0 | 0.1 | 0.0 | 0.4 | 89.1 | 1.9 | 0.1 |
| 18 | 2.3 | 0.1 | 1.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.1 | 0.0 | 2.0 | 0.1 | 0.1 | 0.5 | 0.0 | 0.2 | 0.1 | 0.4 | 3.2 | 89.1 | 0.1 |
| 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.9 | 0.0 | 0.9 | 4.5 | 91.0 |

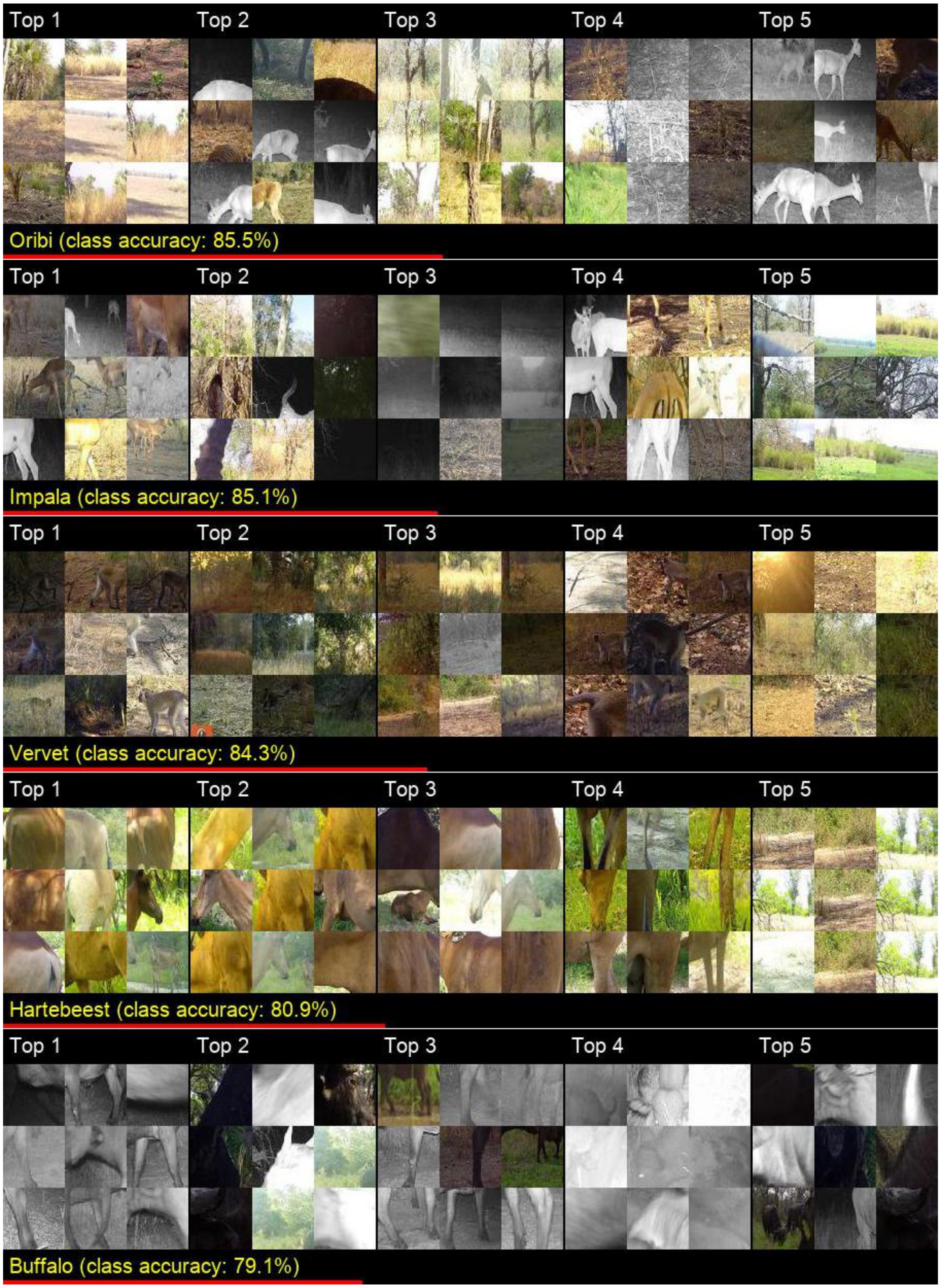
## Mutual information results of all animal species


Top 1
Top 2
Top 3
Top 4
Top 5
Civet (class accuracy: 95.2%)
Top 1
Top 2
Top 3
Top 4
Top 5
Wildebeest (class accuracy: 91.0%)
Top 1
Top 2
Top 3
Top 4
Top 5
Baboon (class accuracy: 90.2%)
Top 1
Top 2
Top 3
Top 4
Top 5
Elephant (class accuracy: 89.6%)
Top 1
Top 2
Top 3
Top 4
Top 5
Porcupine (class accuracy: 89.2%)

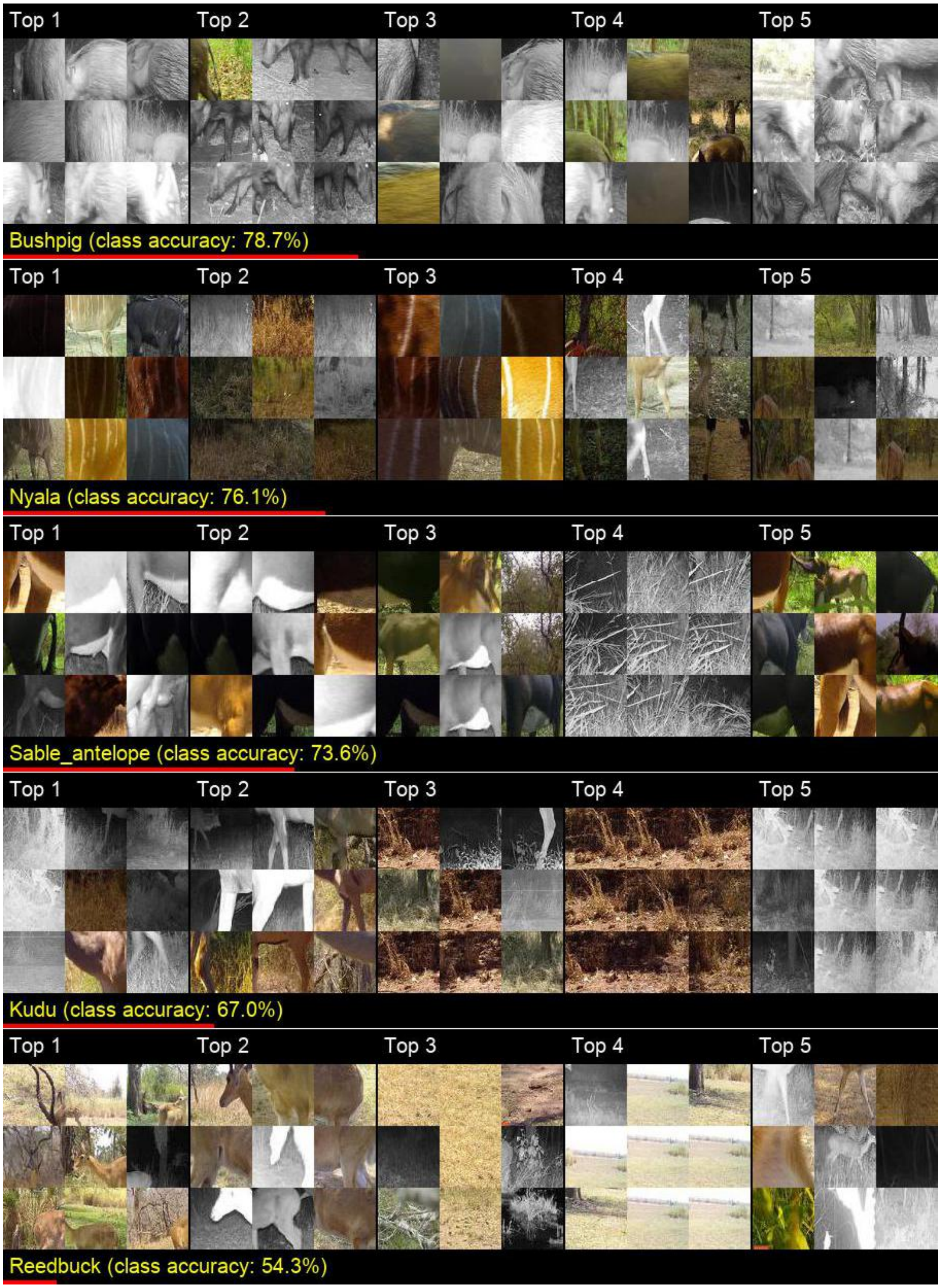
(a)


Top 1
Top 2
Top 3
Top 4
Top 5
Warthog (class accuracy: 89.1%)
Top 1
Top 2
Top 3
Top 4
Top 5
Waterbuck (class accuracy: 89.1%)
Top 1
Top 2
Top 3
Top 4
Top 5
Genet (class accuracy: 88.6%)
Top 1
Top 2
Top 3
Top 4
Top 5
Bushbuck (class accuracy: 86.4%)
Top 1
Top 2
Top 3
Top 4
Top 5
Hare (class accuracy: 86.2%)

(b)


Top 1
Top 2
Top 3
Top 4
Top 5
Oribi (class accuracy: 85.5%)
Top 1
Top 2
Top 3
Top 4
Top 5
Impala (class accuracy: 85.1%)
Top 1
Top 2
Top 3
Top 4
Top 5
Vervet (class accuracy: 84.3%)
Top 1
Top 2
Top 3
Top 4
Top 5
Hartebeest (class accuracy: 80.9%)
Top 1
Top 2
Top 3
Top 4
Top 5
Buffalo (class accuracy: 79.1%)

(c)


Top 1
Top 2
Top 3
Top 4
Top 5
Bushpig (class accuracy: 78.7%)
Top 1
Top 2
Top 3
Top 4
Top 5
Nyala (class accuracy: 76.1%)
Top 1
Top 2
Top 3
Top 4
Top 5
Sable_antelope (class accuracy: 73.6%)
Top 1
Top 2
Top 3
Top 4
Top 5
Kudu (class accuracy: 67.0%)
Top 1
Top 2
Top 3
Top 4
Top 5
Reedbuck (class accuracy: 54.3%)

(d)

Figure A.1: **Common within-species features for all 20 species.** Following Figure 1.7 in Chapter 1 for the case of `Porcupine` and `Reedbuck`, here the extracted patches are centered around the hottest pixel of the 5 most responsive neurons in the last convolutional layer of our CNN that had the highest MI score (Methods) for all 20 species.

## Dice similarity between extracted features and visual descriptors

Figure A.2 shows the DSC scores of all the 20 species, where the mean DSC across species was 0.69 with a standard deviation of 0.13. The closer the score is to 1, the more similar the extracted features are to the human visual descriptors.

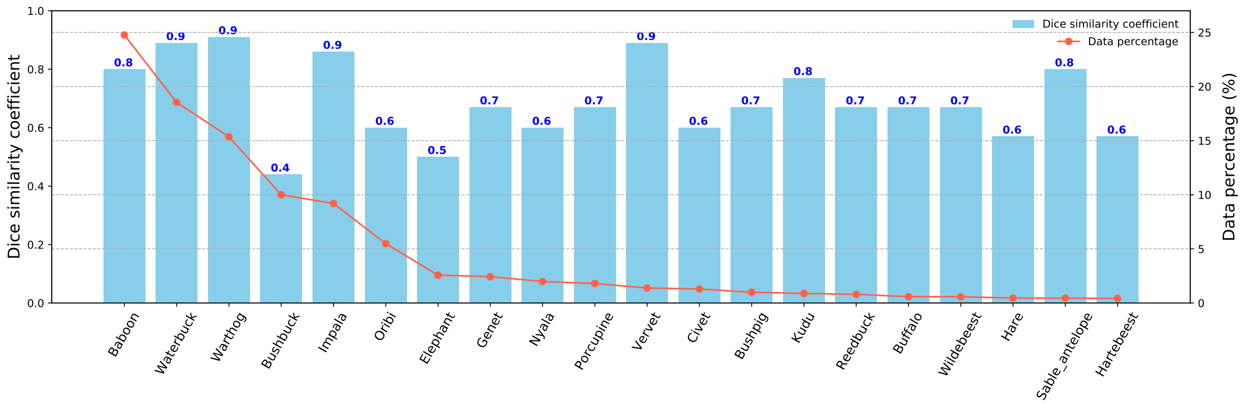


Figure A.2: **Similarities between extracted features and corresponding visual descroptors mostly used by human experts.** The extracted features were agreed upon by up to 4 experts (ZM, KMG, ZL, and MSN), who scored 9 randomly selected images of each species. The similarities were calculated using the DSC between extracted features and human visual features. The scores seem to have no relationship with the class distribution.

## Full visual descriptors

Table A.3: **Full visual descriptors used by humans to identify the 20 most common species from camera trap images.** These features were identified through a survey of people with extensive experience in classifying camera trap data from Gorongosa. The features below were selected by at least 5 of the 13 survey respondents.

| | |
|---|---|
| Baboon | primate body type<br>tail curving upward at base<br>long, dark snout |
| Buffalo | horns that curve to the side of head<br>stocky barrel-shaped body<br>dark coat |
| Bushbuck | thick ring of short, dark fur along neck<br>parallel, slightly spiraled horns<br>rounded rump<br>ungulate body type<br>white spots along the rump |
| Bushpig | pig body type<br>silver-colored mane |
| Civet | nocturnal<br>small carnivore body type<br>black spots<br>rounded back<br>short, black legs<br>crest of black hair from head to tail |
| Elephant | stocky, rectangular body shape<br>gray to brownish wrinkled skin |

| | |
|---|---|
| | long trunk<br>huge ears that are wide at base and narrow at bottom<br>thick, round legs<br>white tusks |
| Genet | slender body<br>long, narrow tail<br>banded tail<br>black spots<br>small carnivore body type |
| Hare | round body<br>long ears that point up |
| Hartebeest | curved horns<br>ungulate body type<br>uniform dark brown coat |
| Impala | S-shaped horns of male<br>tri-colored body<br>black streaks on the rear |
| Kudu | long horns with large spirals (males)<br>hump on back of neck<br>thin, white stripes on back<br>white band between the eyes<br>light gray/brown color<br>ungulate body type<br>long, slender legs |
| Nyala | thin, white stripes on back<br>golden fur of female, dark brown fur of male<br>white spots on face and nose of male |

| | |
|---|---|
| | spiral horns of male<br>ungulate body<br>white and yellow leg markings |
| Oribi | short, straight horns of male<br>white abdomen<br>short, black tail<br>black circular patches under ears<br>conical head shape<br>ungulate body type |
| Porcupine | nocturnal<br>long black and white quills<br>stout, rounded body shape |
| Reedbuck | forward-curving horns of male<br>black circular patches under ears<br>uniform coloration<br>ungulate body shape |
| Sable antelope | long, backward-curving horns<br>horse-like body type<br>white striped facial markings<br>white underbelly<br>chestnut coat of female and dark brown color of male |
| Vervet | primate body type<br>black face<br>long tail, held out straight<br>white brow |
| Warthog | pig body type<br>two pairs of upward-pointing tusks |

| | |
|---|---|
| | mane from top of head to middle of back |
| | thin tail with tuft of hair at the bottom |
| | flat, wide snout |
| Waterbuck | ribbed horns, curved out and forward (male) |
| | white circular ring of fur on rump |
| | shaggy, coarse, red-brown fur |
| | black nose |
| | ungulate body type |
| Wildebeest | curved horns that are wider than they are tall |
| | horse-like body type |
| | long, rectangular face |
| | black beard |
| | black mane along back |
| | black vertical stripes on neck |

# Appendix B: Supplementary material for Chapter 2

## Data

### Data background

Imagery for this study was collected by the U.S. Fish and Wildlife Service at Nantucket Shoals (Cape Cod), Massachusetts, in February 2017 and Lake Michigan near Manitowoc, Michigan, USA, in October 2016. Pixel resolution for the Nantucket Shoals dataset ranged from 0.18 to 1.47 cm and 0.14 to 0.32 cm for the Lake Michigan dataset. The average image dimension of the Cape Cod dataset is 75×79, and the average image dimension of the Lake Michigan dataset is 91×108.

### Data pre-processing

We used cropped images of individual birds for the experiments in this project. The images were manually cropped and annotated by human experts. There are 10,682 cropped images in the Cape Cod dataset and 236 images in the Lake Michigan dataset. There are six different classes (Figure 2.2), one of which is an unknown class only for scoters (`Unknown Scoter`) and a general unknown class (`Non-target Species`) that contain instances for species that were not germane to the initial objectives of the data collection. The overall class distribution is illustrated in Figure 2.2.

We randomly split the Cape Cod dataset into training and testing sets. Detailed numbers of the split are recorded in Table B.1 and Figure 2.2. All the images were resized to 256×256 pixels before being input into the AI models.

Table B.1: **Details of Cape Cod training-testing split, and Lake Michigan testing set.**

| | **Cape Cod** | | **Lake Michigan** |
|---|---|---|---|
| **Species** | **Train #** | **Test #** | **Test #** |
| `Unknown Scoter` | 466 | 114 | - |
| `Black Scoter` | 341 | 108 | - |
| `White-winged Scoter` | 45 | 21 | - |
| `Common Eider` | 6,246 | 3,172 | - |
| `Long-tailed Duck` | 17 | 5 | 231 |
| `Non-target Species` | 108 | 38 | 5 |

# Methods and implementation details

## Classification model

We used ResNet-50 [43], a widely applied Convolutional Neural Network (CNN), as our backbone classification model (i.e., basic classification model without any extra modules for tasks other than pure classification). All the other components for specific experiments were added to this ResNet-50 backbone.

The basic training hyperparameters are recorded in Table B.2. In terms of hyperparameter tuning and validation, we further randomly split the training data into pre-train versus validation sets with a 90%-10% split. All hyperparameters were validated on the 10% validation set. For example, the best number of training epochs used to train the model was obtained when the highest validation performance occurred on the validation set. Once the hyperparameters were validated, we use all the pre-train and validation sets to re-train a model and report our test results on the testing sets with this final model. This way of training produces optimized models that utilize all training data without sacrifices with validation sets and save training efforts from multiple training procedures such as cross-validations.

Table B.2: **List of hyperparameters used in the baseline experiments.**

| Parameters | Values |
|---|---:|
| Baseline architecture | ResNet-50 |
| Starting training epochs | 85 |
| Batch size | 128 |
| Initial learning rate | 0.001 |
| Learning rate decay Epochs | 30 |
| Learning rate decay Ratio | 0.1 |
| Momentum | 0.9 |
| Weight decay | 0.0005 |

## LDAM

As our training dataset was extremely imbalanced (i.e., the most abundant species has 6,246 training samples whereas the least frequent species has only 17 training samples), we added a Label Distribution Aware Marginal (LDAM) training loss function [12] to our model to balance the learning across all species. LDAM is a light-weighted loss re-weighting method for imbalanced recognition (Eq. 3.6). It calculates class-specific margins ($\Delta_y$) based on class sample sizes ($n_y$) to regularize the training focus of each class. The more training instances for a species (i.e., class), the smaller the class-specific margin is, and vice versa.

$$\begin{aligned} \mathcal{L}_{\mathrm{LDAM}}(x,y) &= -\log \frac{e^{f(x)_y - \Delta_y}}{e^{f(x)_y - \Delta_y} + \sum_{j \neq y} e^{f(x)_j}} \\ \Delta_y &= \frac{C}{n_y^{1/4}}, \quad y \in \{1, ..., C\} \end{aligned} \tag{3.6}$$

where $x$ is the input to the classification model. $y$ is corresponding class label. $f(\cdot)$ is the classification model (i.e., ResNet-50 in this project). $\Delta_y$ is the margin of class $y$. $n_y$ is the total number training samples of class $y$. $C$ is total number of classes.

### Soft-fine Pseudo-labels

The next component in our model is a soft-fine label approach that utilizes coarsely annotated images to enhance the generalization of corresponding sub-classes. Specifically, during training, we use `Unknown Scoter` images to help the model learn more discriminative features of scoters and to have better recognition performance on both Black Scoter and White-winged Scoter images.

To implement this soft-fine label approach, we first normalize the outputs of the ResNet-50 model (5 dimension vectors) with a Softmax function. Then we normalize the values that represent `Black Scoter` and `White-winged Scoter` to 1 and set the other three values to 0 (Figure 2.8). Finally, we use these normalized softmax values as our soft-fine labels on `Unknown Scoter` images with an Averaged Binary Cross-entropy (ABCE) loss, a loss function traditionally used for samples with multiple co-occurring labels [132] (Eq. 3.7).

$$\begin{aligned} \mathcal{L}_{\mathrm{ABCE}}(x,y) =& \mathrm{mean}(l_1, ..., l_C) \\ l_n =& -y_n \cdot \log(f(x)_n - \Delta_y) \\ & - (1 - y_n) \cdot \log(1 - (f(x)_n - \Delta_y)) \end{aligned} \tag{3.7}$$

where $y$ is class label. $l_n$ is a binary cross-entropy loss of each class. $\Delta_y$ is the same margin of class $y$ calculated in LDAM loss. $C$ is total number of classes.

The final training loss is a combination of LDAM and soft-fine label ABCE loss:

$$\mathcal{L}_{\mathrm{final}}(x,y) = \mathbb{1}(y \neq \mathrm{SP.}) \cdot \mathcal{L}_{\mathrm{LDAM}}(x,y) + \mu \cdot \mathcal{L}_{\mathrm{ABCE}}(x,y) \tag{3.8}$$

where SP. is super-class.

## Domain adaptation to Lake Michigan

Finally, we have a domain adaptation component that adapts our model trained on Cape Cod data to Lake Michigan data. Because the species composition in Cape Cod and Lake Michigan overlap, the most straightforward adaptation method without complicated components is Semi-supervised learning with pseudo-labels [66] to fine-tune the pre-trained model. In this project, we use FixMatch [117], a state-of-the-art semi-supervised learning method. In FixMatch, the only required extra component is a two-branch data augmentation procedure. One data augmentation is called **weak augmentation**, which only applies random crop and random horizontal flip on the input image. We use the predictions on the weakly augmented images as pseudo-labels. The second data augmentation is **strong augmentation**, which has a random selection of augmentation from a pool of multiple augmentation methods. Table B.3 has all the augmentation selections in our augmentation pool. We use strongly augmented Lake Michigan images with their corresponding pseudo-labels to fine-tune our pre-trained model (Eq. 3.9).

$$\mathcal{L}_{\text{FixMatch}}(x, \hat{y}) = \mathcal{L}_{\text{LDAM}}(\alpha(x),\ \text{argmax}(f(\mathcal{A}(x)))) \tag{3.9}$$

where $\hat{y}$ is pseudo-label. $\alpha(\cdot)$ is weak augmentation. $\mathcal{A}(\cdot)$ is strong augmentation.

Table B.3: **Augmentation pool for FixMatch fine-tuning**

| **Type** | **Pool** |
|---|---|
| **Strong** | Random contrast, color, and brightness enhancement<br>Random color equalizing, posterizing, sharpening, and inversion<br>Random rotating, shearing, fliping, and translate<br>Random image cut out |
| **Weak** | Random flipping |

# Appendix C: Supplementary material for Chapter 3

## Data

### Data background

The camera trap data come from the WildCam Gorongosa long-term research and monitoring program in Gorongosa National Park, Mozambique (18.8154, 34.4963) [36], the same project introduced in Chapter 1. The data used in Chapter 3 were collected from 2016 to 2019. There are 630,544 images in total.

#### Data split and preprocessing

The data set was randomly split into two groups of training and validation sets to mimic periodical data collection from two sequential time periods, along with an additional `Unknown` set for improving and validating the model's sensitivity to novel and difficult samples. To reduce bias, we split the data set based on camera trigger events, such that both images in a paired trigger event were either both in the training or validation set. The training-validation split did not account for camera locations (i.e., images from a given camera were present in both testing and training sets). For large-scale, long-term projects, it is more likely that the camera locations are stable, and in our study, the cameras cover most of the landscapes in the monitoring area and include a diversity of background types that change seasonally throughout the year. Possible distribution shifts in our data set solely come from temporal animal community changes instead of spatial landscape/ecosystem changes.

The first group contains the 26 most abundant categories, and the second period contains all 41 categories. We randomly divided each period into training (80% of samples) and validation (20% of samples) sets. For scarce categories that had fewer than 80 images (e.g., `Crested Guineafowl`, `Eland`, `Lion`, and `Serval`), we randomly selected 20 samples instead of 20% of the data to ensure the quality of validation. The labels and distributions of these two groups of data are illustrated in Figure 3.3.

Within the 14 categories that were tagged `Unknown`, we randomly selected 80% data to fine-tune the model's sensitivity to novel and difficult samples. We then used the rest of the sample from the 14 categories as an extra validation set to evaluate the model's novel image detection capacity.

All of the images used in Chapter 3 were first resized to 256x256. For training inputs, these images were randomly cropped and resized to 224x224. For validation and inference inputs, images were center cropped to 224x224. Table C.1 reports the list of data augmentations used for training and corresponding hyper-parameters.

# Methods and implementation details

In this section, we report the implementation details of our method. It was developed with Python as the programming language and Pytorch [90] as the deep learning framework. The detailed experimental pipeline is illustrated in Figure 3.4.

## Period 1 and baseline model training

There are two steps in this period: 1) baseline model training on the group 1 data, and 2) classifier fine-tuning using the 14 "left-out" categories for better sensitivity to novel and difficult samples.

### Baseline model

We used ResNet-50 [46] as our baseline model. It was pre-trained on ImageNet [27], a generalized object oriented data set for model weight initialization. The pre-trained model was then trained on the group 1 training data with 26 categories. All the hyperparameters can be found in Table C.2. Model weights with the best validation performance on group 1 validation data were saved as the best model.

### Energy-based fine-tuning

After training on group 1 data, we used energy-based loss [71] and the 14 "left-out" categories (tagged as `Unknown`) to fine-tune the classifier for better sensitivity to novel and difficult samples. The energy-based loss was calculated as Eq. 3.10:

$$\begin{aligned} L_{\text{energy}} = &\mathbb{E}_{x_{\text{known}} \sim \mathfrak{D}^{\text{train}}_{\text{known}}} (\max(0, E(x_{\text{known}}) - m_{\text{known}})^2 \\ &+\mathbb{E}_{x_{\text{unknown}} \sim \mathfrak{D}^{\text{train}}_{\text{unknown}}} (\max(0, m_{\text{unknown}} - E(x_{\text{unknown}}))^2 \end{aligned} \tag{3.10}$$

$$E(x) = -T \cdot \log \sum_{i}^{N} e^{(f(x_i)/T)} \tag{3.11}$$

where $\mathbb{E}$ is expectation, $x_{\text{known}}$ and $x_{\text{unknown}}$ are samples from group 1 and the 14 `Unknown` categories, respectively. $\mathfrak{D}^{\text{train}}_{\text{known}}$ and $\mathfrak{D}^{\text{train}}_{\text{unknown}}$ represents data sets of group 1 and 14 "Unknown" categories. $E(\cdot)$ is *Helmholtz free energy*, calculated as the log sum of outputs from the network. $f(\cdot) : \mathbb{R}^{D \times D} \to \mathbb{R}^{K}$ is the network that maps $D \times D$ images to $K$ dimensional vectors. $T$ is the temperature that regularizes the energy. $m_{\text{known}}$ and $m_{\text{unknown}}$ are two margins applied on known and unknown energy.

During fine-tuning, both cross-entropy loss and energy-based loss are tuned. Eq. 3.12 is the final loss, where $w$ is the weight applied on energy-based loss.

$$L = L_{\text{cross\_entropy}} + w \cdot L_{\text{energy}} \tag{3.12}$$

All hyperparameters are reported in Table C.2.

## Period 2 and model update

### Active selection and confidence calculation

Following [71], prediction confidence for active selection is calculated based on *Helmholtz free energy* (Eq. 3.11). Based on a preset energy threshold $\tau$, predictions are separated into high- and low-confidence. In other words, predictions are considered confident if $-E(x) > \tau$ and vice versa. Based on prediction confidence, low-confidence predictions are assigned human annotations, and high-confidence predictions are utilized as initial pseudo-labels for semi-supervised learning.

### Pseudo-labels and semi-supervised learning

Pseudo-label semi-supervision utilizes both human annotations and pseudo-labels to update the model. In the original approach, where models are randomly initialized, pseudo-labels get updated throughout training iterations [66]. In other words, at each iteration, the model predicts samples without human annotations and uses these predictions as pseudo-labels to train the same samples with a stronger set of data augmentations. In our approach, as the pseudo-labels usually have higher quality than random predictions, we set three semi-update repeats and only updated the pseudo-labels at the beginning of each repeat using the best model from the last repeat. Specifically, within each semi-update repeat, the model was updated with a fixed set of pseudo-labels and a number of training epochs. Model weights with the best validation performance were saved, and at the end of the repeat, the best model was used to predict samples without human annotations to produce a new set of pseudo-labels, and a new repeat started. Only model weights with the best validation performance throughout the three repeats were saved, and the number of repeats is a hyper-parameter that can be tuned using validation data. Other hyper-parameters can also be found in Table C.2.

### OLTR

OLTR is an additional component in our framework targeting the long-tailed distribution of classes in the data sets. Generally speaking, it uses embedding level memory of each category to enhance the distinguishability of scarce categories. It is based on the idea that a lot of the mid-level visual features (i.e., feature embedding) are shared between similar categories (e.g., most of the antelopes share similar body shapes). Since the model can usually learn high quality feature embeddings from abundant species, through memory selection techniques, the model is able to select relevant feature embedding to help improve the distinguishability of scare categories. We directly apply OLTR to our framework. For a detailed explanation of OLTR, please refer to the original paper [74].

Table C.1: **List of the augmentation methods and corresponding parameters we used on our training data.**

| Augmentations | Parameters | Values |
|---|---|---|
| Random resize crop | Dimension | $224 \times 224$ |
| | Range of crop scale | $0.08 \sim 1.0$ |
| | Range of crop aspect ratio | $0.8 \sim 1.2$ |
| Random gray scale | Probability | 0.1 |
| Random horizontal flip | Probability | 0.5 |
| Random rotation | Probability | 0.5 |
| | Rotation degree | 45 |
| Color jittering | Brightness jittering | 0.4 |
| | Contrast jittering | 0.4 |
| | Saturation jittering | 0.4 |
| | Hue jittering | 0.1 |
| Normalization | Mean | $[0.485, 0.456, 0.406]$ |
| | Std | $[0.229, 0.224, 0.225]$ |

Table C.2: **List of hyperparameters of our framework used in the two-period experiments.**

| Period | Parameters | Values |
|---|---|---|
| Period 1. Training | Baseline architecture | ResNet-50 |
| | Training epochs | 40 |
| | Batch size | 64 |
| | Initial learning rate (feature) | 0.001 |
| | Initial learning rate (classifier) | 0.01 |
| | Learning rate decay Epochs | 10 |
| | Learning rate decay Ratio | 0.1 |
| | Momentum | 0.9 |
| | Weight decay | 0.0005 |
| Period 1. Energy Fine-tuning | Training epochs | 10 |
| | Batch size | 96 |
| | Known : Unknown ratio | 1:2 |
| | Energy loss weight | 0.01 |
| | Initial learning rate (feature) | 0.00001 |
| | Initial learning rate (classifier) | 0.0001 |
| | Confidence threshold ($\tau$) | 13.7 |
| | Energy temperature | 1.5 |
| Period 2. Updating | Baseline architecture | ResNet-50 + OLTR |
| | Semi-repeats | 3 |
| | Epochs in each repeat | 30 |
| | Batch size | 64 |
| | Pseudo-label % | 50% |
| | Initial learning rate of each repeat (feature) | 0.0001 |
| | Initial learning rate of each repeat (classifier) | 0.01 |
| | Initial learning rate of each repeat (memory) | 0.0001 |
| | Learning rate decay Epochs | 10 |
| | Learning rate decay Ratio | 0.1 |
| | Momentum | 0.9 |
| | Weight decay | 0.0005 |
| Period 2. Energy Fine-tuning | Training epochs | 10 |
| | Batch size | 96 |
| | Known : Unknown ratio | 1:2 |
| | Energy loss weight | 0.01 |
| | Initial learning rate (feature) | 0.000001 |
| | Initial learning rate (classifier) | 0.00001 |
| | Initial learning rate (memory) | 0.000001 |
| | Confidence threshold ($\tau$) | 6.7 |
| | Energy temperature | 0.06 |

# Additional results

We report detailed results of model update performance by category in Table C.3.

Table C.3: **Classification performance comparisons of Period 2 by category between our method and fully annotated transfer learning.**

| | | Traditional transfer learnig w/ full human ann. | | Our framework (Semi-OLTR) | |
|---|---|---|---|---|---|
| | **Species** | **# of Human Ann.** | **Acc. (%)** | **# of Human Ann.** | **Acc. (%)** |
| Exist in $Group_{1\&2}$ | Ghost | 20500 | 96.2 | 4248 | 90.2 |
| | Waterbuck | 17938 | 88.8 | 2079 | 82.4 |
| | Baboon | 15660 | 87.3 | 2335 | 81.1 |
| | Warthog | 17400 | 87.4 | 4224 | 79.7 |
| | Bushbuck | 6622 | 84.5 | 2179 | 72.3 |
| | Impala | 7153 | 84.0 | 1306 | 77.1 |
| | Oribi | 3832 | 83.8 | 966 | 76.7 |
| | Elephant | 2471 | 88.2 | 470 | 85.1 |
| | Genet | 1976 | 85.5 | 888 | 84.0 |
| | Nyala | 1569 | 73.9 | 434 | 75.1 |
| | Setup | 1229 | 87.4 | 389 | 86.0 |
| | Bushpig | 1040 | 83.1 | 377 | 83.1 |
| | Porcupine | 1152 | 83.9 | 300 | 88.3 |
| | Civet | 699 | 82.9 | 123 | 83.9 |
| | Vervet | 739 | 73.2 | 263 | 81.0 |
| | Reedbuck | 740 | 65.8 | 203 | 75.3 |
| | Kudu | 556 | 70.9 | 161 | 77.2 |
| | Buffalo | 479 | 89.0 | 63 | 84.8 |
| | Sable_antelope | 323 | 85.2 | 48 | 86.1 |
| | Duiker_red | 370 | 86.8 | 116 | 89.6 |
| | Hartebeest | 394 | 91.2 | 63 | 84.6 |
| | Wildebeest | 303 | 83.5 | 44 | 82.4 |
| | Guineafowl_helmeted | 304 | 64.6 | 250 | 74.4 |
| | Hare | 214 | 78.8 | 166 | 80.8 |
| | Duiker_common | 194 | 62.7 | 92 | 80.4 |
| | Fire | 160 | 100.0 | 14 | 100.0 |
| Exist in $Group_2$ Only | Mongoose_marsh | 343 | 70.6 | 287 | 71.8 |
| | Aardvark | 235 | 77.6 | 128 | 81.0 |
| | Honey_badger | 234 | 60.3 | 190 | 63.8 |
| | Hornbill_ground | 203 | 80.0 | 161 | 72.0 |
| | Mongoose_slender | 165 | 68.0 | 157 | 72.0 |
| | Mongoose_bushy_tailed | 161 | 74.0 | 106 | 72.0 |
| | Samango | 99 | 58.0 | 48 | 70.0 |
| | Mongoose_white_tailed | 84 | 52.0 | 79 | 64.0 |
| | Mongoose_banded | 70 | 38.0 | 62 | 52.0 |
| | Mongoose_large_grey | 63 | 44.0 | 54 | 48.0 |
| | Bushbaby | 39 | 36.0 | 31 | 50.0 |
| | Guineagowl_crested | 46 | 95.0 | 35 | 100.0 |
| | Eland | 44 | 90.0 | 31 | 70.0 |
| | Lion | 42 | 70.0 | 32 | 75.0 |
| | Serval | 41 | 45.0 | 32 | 60.0 |

Red color means higher performance.

# Bibliography

[1] Sharath Adavanne et al. “Stacked convolutional and recurrent neural networks for bird audio detection”. In: *2017 25th European signal processing conference (EUSIPCO)*. IEEE. 2017, pp. 1729–1733.

[2] Jorge A Ahumada et al. “Community structure and diversity of tropical forest mammals: data from a global camera trap network”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 366.1578 (2011), pp. 2703–2711.

[3] Jorge A Ahumada et al. “Wildlife insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet”. In: *Environmental Conservation* 47.1 (2020), pp. 1–6.

[4] T Michael Anderson et al. “The spatial distribution of African savannah herbivores: species associations and habitat occupancy in a landscape context”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 371.1703 (2016), p. 20150314.

[5] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[6] Jos Barlow et al. “Anthropogenic disturbance in tropical forests can double biodiversity loss from deforestation”. In: *Nature* 535.7610 (2016), pp. 144–147.

[7] Anthony D Barnosky et al. “Has the Earth’s sixth mass extinction already arrived?” In: *Nature* 471.7336 (2011), pp. 51–57.

[8] R. Battiti. “Using mutual information for selecting features in supervised neural net learning”. In: *IEEE Transactions on Neural Networks* 5.4 (July 1994), pp. 537–550. ISSN: 1045-9227. DOI: 10.1109/72.298224.

[9] A Cole Burton et al. “Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes”. In: *Journal of Applied Ecology* 52.3 (2015), pp. 675–685.

[10] A. Cole Burton et al. “REVIEW: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes”. In: *Journal of Applied Ecology* 52.3 (2015), pp. 675–685. DOI: 10.1111/1365-2664.12432.

[11] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. “Ace: Ally complementary experts for solving long-tailed recognition in one-shot”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 112–121.

[12] Kaidi Cao et al. “Learning imbalanced data sets with label-distribution-aware margin loss”. In: *arXiv preprint arXiv:1906.07413* (2019).

[13] Anthony Caravaggi et al. “A review of camera trapping for conservation behaviour research”. In: *Remote Sensing in Ecology and Conservation* 3.3 (2017), pp. 109–122. DOI: `10.1002/rse2.48`.

[14] Anthony Caravaggi et al. “An invasive-native mammalian species replacement process captured by camera trap survey random encounter models”. In: *Remote Sensing in Ecology and Conservation* 2.1 (2016), pp. 45–58.

[15] Dominique Chabot and Charles M Francis. “Computer-automated bird detection and counts in high-resolution aerial images: a review”. In: *Journal of Field Ornithology* 87.4 (2016), pp. 343–359.

[16] Robert D Chambers et al. “Deep learning classification of canine behavior using a single collar-mounted accelerometer: Real-world validation”. In: *Animals* 11.6 (2021), p. 1549.

[17] Prithvijit Chattopadhyay et al. “Evaluating visual conversational agents via cooperative human-AI games”. In: *arXiv:1708.05122* (2017).

[18] Guobin Chen et al. “Deep convolutional neural network based species recognition for wild animal monitoring”. In: *2014 IEEE International Conference on Image Processing*. IEEE. 2014, pp. 858–862.

[19] Lauren M Chronister et al. *An annotated set of audio recordings of Eastern North American birds containing frequency, time, and species information*. 2021.

[20] Miguel Clavero and Emili Garcia-Berthou. “Invasive species are a leading cause of animal extinctions”. In: *Trends in ecology & evolution* 20.3 (2005), p. 110.

[21] Evangeline Corcoran et al. “Automated detection of koalas using low-level aerial surveillance and machine learning”. In: *Scientific reports* 9.1 (2019), pp. 1–9.

[22] C Crisci, B Ghattas, and G Perera. “A review of supervised machine learning algorithms and their applications to ecological data”. In: *Ecological Modelling* 240 (2012), pp. 113–122.

[23] Yin Cui et al. “Class-balanced loss based on effective number of samples”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.

[24] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[25] Michele Dalponte, Lorenzo Frizzera, and Damiano Gianelle. “Individual tree crown delineation and tree species classification with hyperspectral and LiDAR data”. In: *PeerJ* 6 (2019), e6227.

[26] J. Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[27] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[28] Jia Deng et al. "Large-scale object classification using label relation graphs". In: *European conference on computer vision*. Springer. 2014, pp. 48–64.

[29] Stig Descamps et al. "An automatic counter for aerial images of aggregations of large birds". In: *Bird study* 58.3 (2011), pp. 302–308.

[30] Terrance DeVries and Graham W Taylor. "Learning Confidence for Out-of-Distribution Detection in Neural Networks". In: *arXiv preprint arXiv:1802.04865* (2018).

[31] Rodolfo Dirzo et al. "Defaunation in the Anthropocene". In: *science* 345.6195 (2014), pp. 401–406.

[32] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874.

[33] Karina Figueroa et al. "Fast automatic detection of wildlife in images from trap cameras". In: *Iberoamerican Congress on Pattern Recognition*. Springer. 2014, pp. 940–947.

[34] Peter C Frederick et al. "Accuracy and variation in estimates of large numbers of birds by individual observers using an aerial survey simulator". In: *Journal of Field Ornithology* 74.3 (2003), pp. 281–287.

[35] Geoffrey French, Michal Mackiewicz, and Mark Fisher. "Self-ensembling for visual domain adaptation". In: *arXiv preprint arXiv:1706.05208* (2017).

[36] K. M. Gaynor et al. "Postwar wildlife recovery in an African savanna: evaluating patterns and drivers of species occupancy and richness". In: *Animal Conservation* (2020). ISSN: 1367-9430. DOI: 10.1111/acv.12661.

[37] Alexander Gomez et al. "Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds". In: *International Symposium on Visual Computing*. Springer. 2016, pp. 747–756.

[38] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[39] Will Grathwohl et al. "Your classifier is secretly an energy based model and you should treat it like one". In: *arXiv preprint arXiv:1912.03263* (2019).

[40] Emilio Guirado et al. "Whale counting in satellite and aerial images with deep learning". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[41] Chuan Guo et al. "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330.

[42] Yann Hautier et al. "Anthropogenic environmental changes affect ecosystem stability via biodiversity". In: *Science* 348.6232 (2015), pp. 336–340.

[43] Haibo He and Edwardo A Garcia. "Learning from imbalanced data". In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1263–1284.

[44] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 770–778. ISSN: 1063-6919. DOI: 10.1109/CVPR.2016.90.

[45] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[46] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[47] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. “Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 41–50.

[48] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[49] Judy Hoffman et al. “Cycada: Cycle-consistent adversarial domain adaptation”. In: *ICML*. 2018.

[50] Suk-Ju Hong et al. “Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery”. In: *Sensors* 19.7 (2019), p. 1651.

[51] Sanaul Hoque, MAHB Azhar, and Farzin Deravi. “ZOOMETRICS-biometric identification of wildlife using natural body marks”. In: *International Journal of Bio-Science and Bio-Technology* 3.3 (2011), pp. 45–53.

[52] Seyedehfaezeh Hosseininoorbin et al. “Deep learning-based cattle behaviour classification using joint time-frequency data representation”. In: *Computers and Electronics in Agriculture* 187 (2021), p. 106241.

[53] Yen-Chang Hsu et al. “Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10951–10960.

[54] Stefan Kahl et al. “BirdNET: A deep learning solution for avian diversity monitoring”. In: *Ecological Informatics* 61 (2021), p. 101236.

[55] Christian Kampichler et al. “Classification in conservation biology: a comparison of five machine-learning methods”. In: *Ecological Informatics* 5.6 (2010), pp. 441–450.

[56] Bingyi Kang et al. “Decoupling representation and classifier for long-tailed recognition”. In: *arXiv preprint arXiv:1910.09217* (2019).

[57] Roland Kays, William J McShea, and Martin Wikelski. “Born-digital biodiversity data: Millions and billions”. In: *Diversity and Distributions* 26.5 (2020), pp. 644–648.

[58] Roland Kays et al. “An empirical evaluation of camera trap study design: How many, how long and when?” In: *Methods in Ecology and Evolution* (2020).

[59] Roland Kays et al. “Does hunting or hiking affect wildlife communities in protected areas?” In: *Journal of Applied Ecology* 54.1 (2017), pp. 242–252. DOI: 10.1111/1365-2664.12700.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105.

[61] Hjalmar S Kühl and Tilo Burghardt. “Animal biometrics: quantifying and detecting phenotypic appearance”. In: *Trends in Ecology & Evolution* 28.7 (2013), pp. 432–441.

[62] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015), pp. 1332–1338.

[63] Sebastian Lapuschkin et al. “Unmasking Clever Hans Predictors and Assessing What Machines Really Learn”. In: *Nature Communications* (2019).

[64] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 0018-9219. DOI: `10.1109/5.726791`.

[65] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444. DOI: `10.1038/nature14539`. URL: `https://doi.org/10.1038/nature14539`.

[66] Dong-Hyun Lee. “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks”. In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. 2013.

[67] Shiyu Liang, Yixuan Li, and R Srikant. “Enhancing the reliability of out-of-distribution image detection in neural networks”. In: *ICLR*. 2018.

[68] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[69] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[70] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[71] Weitang Liu et al. “Energy-based Out-of-distribution Detection”. In: *Advances in Neural Information Processing Systems* (2020).

[72] Yang Liu et al. “Performance comparison of deep learning techniques for recognizing birds in aerial images”. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE. 2018, pp. 317–324.

[73] Z. Liu et al. “Deep Learning Face Attributes in the Wild”. In: *2015 IEEE International Conference on Computer Vision*. Dec. 2015, pp. 3730–3738. DOI: `10.1109/ICCV.2015.425`.

[74] Ziwei Liu et al. “Large-scale long-tailed recognition in an open world”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2537–2546.

[75] Ziwei Liu et al. “Open Compound Domain Adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12406–12415.

[76] David G Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[77] Tim C. D. Lucas et al. "A generalised random encounter model for estimating animal density with remote sensor data". In: *Methods in Ecology and Evolution* 6.5 (2015), pp. 500–509. DOI: 10.1111/2041-210X.12346. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12346. URL: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.12346.

[78] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002. ISBN: 0521642981.

[79] J. Marcus Rowcliffe. "Key frontiers in camera trapping research". In: *Remote Sensing in Ecology and Conservation* 3.3 (2017), pp. 107–108. DOI: 10.1002/rse2.65.

[80] John F McEvoy, Graham P Hall, and Paul G McDonald. "Evaluation of unmanned aerial vehicle shape, flight path and camera type for waterfowl surveys: disturbance effects and species recognition". In: *PeerJ* 4 (2016), e1831.

[81] L David Mech et al. "Gray Wolf (Canis lupus) recolonization failure: a Minnesota case study". In: *The Canadian Field-Naturalist* 133.1 (2019), pp. 60–65.

[82] Zhongqi Miao et al. "Insights and approaches using deep learning to classify wildlife". In: *Scientific reports* 9.1 (2019), pp. 1–9.

[83] Zhongqi Miao et al. *Iterative Human and Automated Identification of Wildlife Images*. 2021. arXiv: 2105.02320 [cs.CV].

[84] Fionn Murtagh and Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion?" In: *Journal of Classification* 31.3 (Oct. 2014), pp. 274–295. ISSN: 1432-1343. DOI: 10.1007/s00357-014-9161-z. URL: https://doi.org/10.1007/s00357-014-9161-z.

[85] Mohammad Sadegh Norouzzadeh et al. "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning". In: *Proceedings of the National Academy of Sciences* (2018). ISSN: 0027-8424. DOI: 10.1073/pnas.1719367115. eprint: http://www.pnas.org/content/early/2018/06/04/1719367115.full.pdf. URL: http://www.pnas.org/content/early/2018/06/04/1719367115.

[86] Mohammad Sadegh Norouzzadeh et al. "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning". In: *Proceedings of the National Academy of Sciences* 115.25 (2018), E5716–E5725. ISSN: 0027-8424. DOI: 10.1073/pnas.1719367115. eprint: https://www.pnas.org/content/115/25E5716.full.pdf. URL: https://www.pnas.org/content/115/25/E5716.

[87] Allan F O'Connell, James D Nichols, and K Ullas Karanth. *Camera traps in animal ecology: methods and analyses*. Springer Science & Business Media, 2010.

[88] MS Palmer et al. "A 'dynamic'landscape of fear: prey responses to spatiotemporal variations in predation risk across the lunar cycle". In: *Ecology Letters* 20.11 (2017), pp. 1364–1373.

[89] Adam Paszke et al. "Automatic differentiation in PyTorch". In: 2017.

[90] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[91] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), pp. 1226–1238.

[92] Xingchao Peng et al. "Moment matching for multi-source domain adaptation". In: *ICCV*. 2019.

[93] Stuart L Pimm et al. "The biodiversity of species and their rates of extinction, distribution, and protection". In: *science* 344.6187 (2014).

[94] Tomaso Poggio and Fabio Anselmi. *Visual Cortex and Deep Networks: Learning Invariant Representations*. MIT Press, 2016.

[95] David Powers. "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation". In: *Machine Learning Technologies* 2 (Jan. 2008).

[96] Karel Prach and Lawrence R Walker. "Four opportunities for studies of ecological succession". In: *Trends in Ecology & Evolution* 26.3 (2011), pp. 119–123.

[97] Dede Aulia Rahman, Georges Gonzalez, and Stéphane Aulagnier. "Population size, distribution and status of the remote and Critically Endangered Bawean deer Axis kuhlii". In: *Oryx* 51.4 (2017), pp. 665–672. DOI: `10.1017/S0030605316000429`.

[98] Pranav Rajpurkar et al. "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists". In: *PLoS medicine* 15.11 (2018), e1002686.

[99] Thiago Fernando Rangel and Rafael Dias Loyola. "Labeling ecological niche models". In: *Natureza & Conservação* 10.2 (2012), pp. 119–126.

[100] JV Redfern et al. "Biases in estimating population size from an aerial census: A case study in the Kruger National Park, South Africa: Starfield Festschrift". In: *South African Journal of Science* 98.9 (2002), pp. 455–461.

[101] "Reinstating a landscape of fear: community behavioral responses to predator reintroduction". In: ().

[102] Lindsey N Rich et al. "Assessing global patterns in mammalian carnivore occupancy and richness by integrating local camera trap surveys". In: *Global Ecology and Biogeography* 26.8 (2017), pp. 918–929.

[103] Lindsey N. Rich et al. "Using camera trapping and hierarchical occupancy modelling to evaluate the spatial ecology of an African mammal community". In: *Journal of Applied Ecology* 53.4 (2016), pp. 1225–1235. DOI: `10.1111/1365-2664.12650`.

[104] William J Ripple et al. "Conserving the world's megafauna and biodiversity: The fierce urgency of now". In: *Bioscience* 67.3 (2017), pp. 197–200.

[105] Lior Rokach and Oded Maimon. “Clustering methods”. In: *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 321–352.

[106] Kate Saenko et al. “Adapting visual category models to new domains”. In: *ECCV*. 2010.

[107] D Blake Sasse. “Job-related mortality of wildlife workers in the United States, 1937-2000”. In: *Wildlife society bulletin* (2003), pp. 1015–1020.

[108] Walter J Scheirer et al. “Toward open set recognition”. In: *TPAMI* (2013).

[109] Johannes L Schonberger et al. “Comparative evaluation of hand-crafted and learned local features”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1482–1491.

[110] Marco Seeland et al. “Image-based classification of plant genus and family for trained and untrained plant species”. In: *BMC Bioinformatics* 20.1 (Jan. 2019), p. 4. ISSN: 1471-2105. DOI: `10.1186/s12859-018-2474-x`. URL: `https://doi.org/10.1186/s12859-018-2474-x`.

[111] R R Selvaraju et al. “Grad-CAM: visual explanations from deep networks via gradient-based localization”. In: *2017 IEEE International Conference on Computer Vision* (2017), pp. 618–626. ISSN: 2380-7504. DOI: `10.1109/ICCV.2017.74`.

[112] Léonard Seydoux et al. “Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning”. In: *Nature communications* 11.1 (2020), pp. 1–12.

[113] Li Shen, Zhouchen Lin, and Qingming Huang. “Relay backpropagation for effective learning of deep convolutional neural networks”. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 467–482.

[114] Shoaib Ahmed Siddiqui et al. “Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data”. In: *ICES Journal of Marine Science* 75.1 (2018), pp. 374–389. DOI: `10.1093/icesjms/fsx109`.

[115] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[116] “Snapshot Safari: a large-scale collaborative to monitor Africa’s remarkable biodiversity”. In: *South Africa Journal of Science* (2021).

[117] Kihyuk Sohn et al. “Fixmatch: Simplifying semi-supervised learning with consistency and confidence”. In: *arXiv preprint arXiv:2001.07685* (2020).

[118] Th A Sorensen. “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons”. In: *Biol. Skar.* 5 (1948), pp. 1–34.

[119] Jost Tobias Springenberg et al. “Striving for simplicity: the all convolutional net”. In: *arXiv preprint arXiv:1412.6806* (2014).

[120] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[121] Robin Steenweg et al. “Scaling-up camera traps: Monitoring the planet’s biodiversity with networks of remote sensors”. In: *Frontiers in Ecology and the Environment* 15.1 (2017), pp. 26–34.

[122] Robin Steenweg et al. “Scaling-up camera traps: monitoring the planet’s biodiversity with networks of remote sensors”. In: *Frontiers in Ecology and the Environment* 15.1 (2017), pp. 26–34. DOI: `10.1002/fee.1448`.

[123] Xin Sun et al. “Transferring deep knowledge for object recognition in Low-quality underwater videos”. In: *Neurocomputing* 275 (2018), pp. 897–908. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2017.09.044`. URL: `http://www.sciencedirect.com/science/article/pii/S0925231217315631`.

[124] Alexandra Swanson et al. “Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna”. In: *Scientific Data* 2 (2015), 150026 EP -.

[125] Kristijn RR Swinnen et al. “A novel method to reduce time investment when processing videos from camera trap studies”. In: *PloS one* 9.6 (2014), e98881.

[126] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[127] Michael A Tabak et al. “Machine learning to classify animal species in camera trap images: Applications in ecology”. In: *Methods in Ecology and Evolution* 10.4 (2019), pp. 585–590.

[128] Michael A Tabak et al. “Machine learning to classify animal species in camera trap images: applications in ecology”. In: *bioRxiv* (2018). DOI: `10.1101/346809`. eprint: `https://www.biorxiv.org/content/early/2018/07/09/346809.full.pdf`. URL: `https://www.biorxiv.org/content/early/2018/07/09/346809`.

[129] Yaniv Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2014, pp. 1701–1708. ISBN: 978-1-4799-5118-5. DOI: `10.1109/CVPR.2014.220`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909616`.

[130] Gemma Taylor et al. “Is reintroduction biology an effective applied science?” In: *Trends in Ecology & Evolution* 32.11 (2017), pp. 873–880.

[131] A. Torralba and A. A. Efros. “Unbiased look at data set bias”. In: *CVPR 2011*. June 2011, pp. 1521–1528. DOI: `10.1109/CVPR.2011.5995347`.

[132] Grigorios Tsoumakas and Ioannis Katakis. “Multi-label classification: An overview”. In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007), pp. 1–13.

[133] Devis Tuia et al. “Perspectives in machine learning for wildlife conservation”. In: *Nature Communications* 13.1 (2022), p. 792. DOI: `10.1038/s41467-022-27980-y`. URL: `https://doi.org/10.1038/s41467-022-27980-y`.

[134] Eric Tzeng et al. “Adversarial discriminative domain adaptation”. In: *CVPR*. 2017.

[135] Grant Van Horn and Pietro Perona. "The Devil is in the Tails: Fine-grained Classification in the Wild". In: *arXiv preprint arXiv:1709.01450* (2017).

[136] Hemanth Venkateswara et al. "Deep hashing network for unsupervised domain adaptation". In: *CVPR*. 2017.

[137] Alexander Gomez Villa, Augusto Salazar, and Francisco Vargas. "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks". In: *Ecological Informatics* 41 (2017), pp. 24–32. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2017.07.004.

[138] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in Neural Information Processing Systems 30*. 2016, pp. 3630–3638.

[139] Michele Volpi and Devis Tuia. "Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images". In: *ISPRS journal of photogrammetry and remote sensing* 144 (2018), pp. 48–60.

[140] Jana Wäldchen and Patrick Mäder. "Machine learning for image based species identification". In: *Methods in Ecology and Evolution* (2018), pp. 1–10.

[141] Jana Wäldchen et al. "Automated plant species identification—Trends and future directions". In: *PLOS Computational Biology* 14.4 (Apr. 2018), pp. 1–19. DOI: 10.1371/journal.pcbi.1005993. URL: https://doi.org/10.1371/journal.pcbi.1005993.

[142] Dongliang Wang, Quanqin Shao, and Huanyin Yue. "Surveying wild animals from satellites, manned aircraft and unmanned aerial systems (UASs): a review". In: *Remote Sensing* 11.11 (2019), p. 1308.

[143] Xudong Wang et al. "Long-tailed recognition by routing diverse distribution-aware experts". In: *arXiv preprint arXiv:2010.01809* (2020).

[144] Ben G. Weinstein. "A computer vision for animal ecology". In: *Journal of Animal Ecology* 87.3 (2017), pp. 533–545. DOI: 10.1111/1365-2656.12780.

[145] Robin C Whytock et al. "Robust ecological analysis of camera trap data labelled by a machine learning model". In: *Methods in Ecology and Evolution* (2021).

[146] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10687–10698.

[147] Qizhe Xie et al. "Unsupervised data augmentation for consistency training". In: *arXiv preprint arXiv:1904.12848* (2019).

[148] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Proceedings of the 27th International Conference on Neural Information Processing System*. Vol. 2. 2014, pp. 3320–3328.

[149] Xiaoyuan Yu et al. "Automated identification of animal species in camera trap images". In: *EURASIP Journal on Image and Video Processing* 2013.1 (2013), pp. 1–10.

[150] Matthew D. Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 818–833.

[151] Quan-Shi Zhang and Song-Chun Zhu. “Visual interpretability for deep learning: a survey”. In: *Frontiers of Information Technology & Electronic Engineering* 19.1 (2018), pp. 27–39.

[152] B. Zhou et al. “Interpreting Deep Visual Representations via Network Dissection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2858759.

[153] Bolei Zhou et al. “Object detectors emerge in deep scene cnns”. In: *arXiv preprint arXiv:1412.6856* (2014).

[154] Boyan Zhou et al. “Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9719–9728.

[155] Xiaojin Zhu and Andrew B Goldberg. “Introduction to semi-supervised learning”. In: *Synthesis lectures on artificial intelligence and machine learning* 3.1 (2009), pp. 1–130.