ANIMAL DETECTION FOR PHOTOGRAPHIC CENSUSING

Jason Remington Parham

Submitted in Partial Fullfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:
Dr. Charles Stewart, Chair
Dr. Barbara Cutler
Dr. Bülent Yener
Dr. Richard Radke
Dr. Tanya Berger-Wolf

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York

[December 2021]
Submitted November 2021

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

This dissertation is a product of patience. The highest amounts of mental and physical endurance have been given by my wife, Lindsay. Her skill in raising our children, Heidi and Lincoln, has been awe-inspiring, and I thank her sincerely for her perseverance. It can be challenging to explain to young children why work can be so important – important enough to miss dinners or playtimes at the park. I believe and hope that someday they will understand that the missed time together was instead invested into a higher, more urgent obligation. I dedicate this work to them, as they will be the truest beneficiaries of any success my work may find in the pursuit of wildlife conservation.

I also thank my advisor, Dr. Charles Stewart, who has been a compassionate guide in my academic career and in life as a young husband and father. I appreciate his patience and expertise and the guidance from my committee. The generosity and flexibility of my employers during this Ph.D. process is something that I'm not sure I completely understand; I thank Drs. Anthony Hoogs, Matt Turek, Rusty Blue, and Keith Fieldhouse at Kitware, along with Jason Holmberg and Dr. Tanya Berger-Wolf with Wild Me. I also thank the Gordon and Betty Moore Foundation for their financial support. My graduate lab partners at RPI, Drs. Jon Crall and Hendrik Weideman, provided excellent discussions and stimulation on the latest machine learning methods, and I thank Dr. Barbara Cutler for her tranquility in indulging our energy. Jon, if you ever want to camp in the African bush, just let me know. I also thank my peers in machine learning for animal conservation, Sara Beery and Dr. Stephan Schneider, for their work in kindling a small but passionate research community. Drs. Dan Rubenstein, Kaia Tombak, and Megan McSherry have also been instrumental in facilitating this research, and I thank them for their diligence in working with me over the years. I also cannot forget the dedication and benevolence of the research staff at the Ol Pejeta and Lewa conservancies, the Great Grévy's Trust, the Kenya Wildlife Service, and numerous Wildbook projects.

Lastly, I would like to thank my parents, Anthony, Grace, Linda, Harlon Jr., Kent, and Julie, and my siblings Stephany, Harlon III, Joyce, Kelsey, Kyle, and Chad for their continued support. I also appreciate my co-workers at Wild Me, Jon Van Oast, Drew Blount, Colin Kingen, Mark Fisher, Ben Schiener, and Tanya Stere for permitting my chaos and giving me a fulfilling place to work with friends. I also thank Drew and Olga Moskvyak for their work on new detection components and Tanya as honorary editor. Specific thanks to my sisters-in-law Brittany and Kelsey Sundman, and to Ben and Kaia, for their last-minute help, looking at some zebras when nobody else really wanted to.

ABSTRACT

Animal population monitoring is hard to do at large scales. It is too logistically demanding to track thousands of animals with invasive tools like ear tagging, and methods like aerial surveys and hand-based counts cannot track individuals over time. A database of unique animals and their sightings can be a critical tool for conservation; ecologists gain a more intimate and timely understanding of an endangered species' health when they can estimate life expectancy, visualize migration patterns, and quickly measure the effects of conservation policies.

This dissertation proposes photographic censusing, a way to visually track the population of an entire species with as little human effort as possible. The method is based on a two-day event called a photographic censusing rally, formed as a sight-resight study (building off of mark-recapture) to estimate the size of the population. Photographic censusing is highly automated, is designed to be bootstrapable for new species, and uses citizen scientists to collect large volumes of photographs across a large geographic area. A novel 5-component animal detection pipeline is proposed to analyze collected images of animals and filter sightings of animals for ID. The pipeline offers a whole-image classifier for quick filtering, a bounding box localizer to find annotations, an annotation labeler to determine species and viewpoints, a coarse segmentation algorithm to mask the background, and a component to recognize poor sightings, and is evaluated on new datasets.

This research also presents the Great Grévy's Rally (GGR) results from 2016 and 2018. These censusing events attempted to catalog the entire resident population of Grévy's zebra (*Equus grevyi*) in Kenya and, combined, collected over 90,000 images from more than 350 volunteers. The GGR analysis in 2018 was done with automated tools but still required large amounts of work (18,500 human decisions), cost USD $50,000+, and took over three months. This dissertation discusses the work needed during a photographic census and analyzes failure modes that require human interaction. The novel concept of Census Annotation (CA) is introduced to find comparable regions of animals for automated ID, which drastically increases automation. The 56,588 images from GGR 2018 were reprocessed with the latest recommended methods presented in this work; 11,916 annotations were automatically found for comparable, right-side Grévy's zebra; ID curation used 1,297 human decisions before converging, and 2,820$\pm$167 Grévy's zebra were estimated to be in Kenya in 2018. This result is consistent (within 0.3% of the original estimate of 2,812$\pm$171) with previous estimates on GGR 2018 data and was achieved with a 93% reduction in human effort.

CHAPTER 1
INTRODUCTION

How many Grévy's zebra are in Kenya?

The Grévy's zebra (*Equus grevyi*), as seen in Figure 1.1, was last assessed in 2016 as *Endangered* by the IUCN Red List[1]. This crucial designation marks the estimated probability of extinction for this species at 20% (or above) over the next five generations. The population in the late 1980s was estimated to be around 5,800 animals, whereas the population today is believed to be about half that number [1]. As with all species, the apparent health of the Grévy's zebra population is linked to its total number of members. Agonizingly, however, their population numbers have not been tracked closely or consistently within Kenya, their primary country of residence. This lack of clarity presents a literal existential challenge for conservationists, where having access to a reliable, species-level population estimate is foundational to evaluating the impact of conservation policy and monitoring the growth or decline of the species.

To better track its overall health, we would ideally like to perform a *census* of the entire species and be able to do it routinely. A census is distinct from a simple count as the former tracks *individual* animals over time[2]. For example, a census allows researchers to estimate the population size, but it also gathers data that can be used to answer important ecological questions like, "*where are the animals migrating to and from?*", "*what areas are isolated by geography or human development?*", "*who are the members of an animal's social group, and do those groups change?*" or a question as basic as "*what is the average life expectancy in the wild?*" These questions are hard to answer if we only count and record the number of animals seen at a given place and time [3]–[9]. Anonymity is what limits the usefulness of animal population monitoring. A regular census of known individuals allows for a more intimate and up-to-date understanding of the animal population, which is critical when a species is threatened.

Biologists have long used physical tagging to track individuals and estimate animal population

[1]IUCN Red List for Grévy's Zebra: `https://www.iucnredlist.org/species/7950/89624491` (Accessed: Oct. 29, 2021).

Portions of this chapter previously appeared as: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

Portions of this chapter previously appeared as: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

[2]This discussion borrows language commonly used for people; for simplicity, it refers to animals as "individuals".

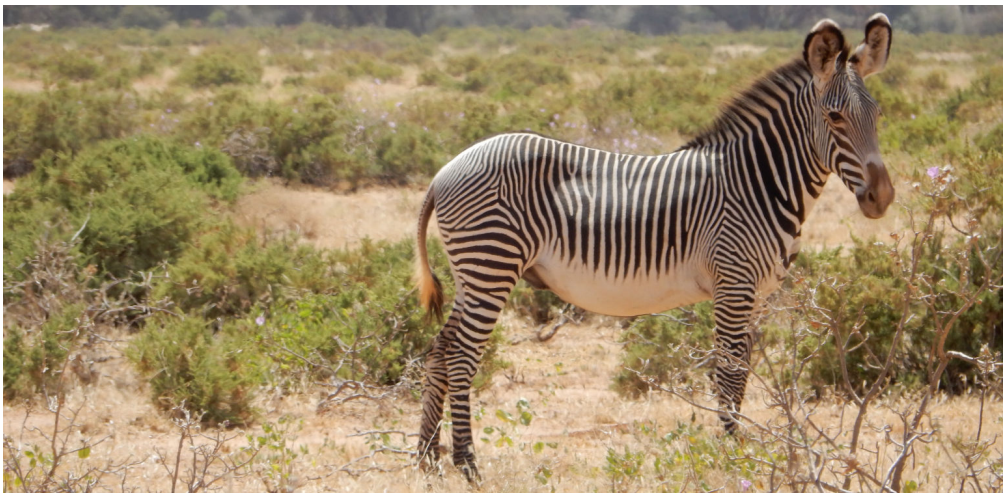


Figure 1.1: An image of a Grévy's zebra in Kenya. Grevy's zebra have thin stripes across the body and a white underbelly. Approximately 90% of the world's Grévy's zebra are located in Kenya [1].

sizes. One of the most popular and prevalent techniques for producing a population size estimate is capture-mark-recapture [10], [11] (or simply "mark-recapture"). Mark-recapture is a sampling technique that starts with an initial capture of the animal population. A second independent capture of the same population is then performed. The number of recaptured animals between the first and second collections is used to estimate the number of animals that were not captured at all. An ecologist may choose to mark the animals from the first capture with paint or use some other type of physical tag to know which animals have been seen before. Performing this kind of detailed mark-recapture study can be prohibitively demanding when the number of individuals in a population grows too large, the population moves across too large of a distance, or the species is difficult to capture due to evasiveness or habitat inaccessibility [12]. Moreover, marking with physical tagging methods like ear tags, metal bands, ear notches, skin branding, or GPS collars can be unreasonably invasive, laborious, expensive, or alter the animal's behavior. These challenges limit how often mark-recapture studies are performed and how comprehensively they can sample the population.

Instead, we would prefer to leverage an animal's intrinsic appearance to "mark" if it has been seen before, taking advantage of a faster and more passive capture process: sight. A better version of mark-recapture can be created that is based on *sights* and *resights* of animals while still relying on the same underlying methodology for estimating the population size. Suppose the visual appearances are unique for each animal in a population and that humans can distinguish two animals based only on their natural markings. In that case, a sight-resight study can be used to track

Table 1.1: A comparison of photographic censusing to existing population estimation methodologies demonstrates that it is better for large animal populations.

Traditional Methods	Photographic Censusing
invasive ear notches, tags, radio collars tranquilizers and veterinary services	**passive** appearance-based with computer vision, does not influence behavior of animals
expensive logistically demanding on research staff and park rangers, requires specialized equipment and training	**inexpensive** distributed, can utilize volunteers and tourists, specialized equipment and training not required
error-prone double counting, human interpretation of data required, tedious verification	**evidence-based** machine learning and computer vision, data-driven decisions, human-in-the-loop verification, statistical estimate
one-time analysis difficult to repeat over time, based on verbal reports cannot audit later	**recurring** allows for tracking individuals over time, ecological trends are available, audit with newer algorithms
Infeasible for large populations	**Ideal for large populations**

encounters of new and known individuals. In short, if we apply this idea to Grévy's zebras, we can think of them almost as walking fingerprints. Just like human thumbprints are unique and linked to a person's identity, the stripe patterns on the side of a zebra are distinctive and unique to that individual. While it is clear that appearance-based ID will not work universally for all species (e.g., a brown squirrel or the skin of a sleek, grey dolphin), we can expect that a giraffe's unique blanket of brown patches, or the intricate layout of scutes on a sea turtle's flipper, or the jagged outline of a whale fluke can be used to recognize and distinguish unique animals.

Expecting a person to remember and recognize hundreds – let alone thousands – of individuals by sight is unrealistic. Therefore, some method of cataloging is needed to help keep records of the animals that have been seen. As the catalog grows, however, the amount of work required to keep tabs on an ever-increasing number of animals can become too demanding or error-prone. Doing this work by hand with notes or physical pictures is therefore not a scalable solution. One straightforward option is to use computational aides that can help store, sort, retrieve, compare, and curate a digital catalog (i.e., a database) of the different encountered animals. Digital photographs are ideal when building such an appearance-based digital catalog of animal IDs as they are easy to

capture and store. Furthermore, taking a photograph of an animal allows for the passive collection of its identifying information and provides a piece of evidence for who, when, and where that individual was seen. A digital image is convenient not only because humans or algorithms can review it to build the database, but it can also be retrieved and re-examined in the future. An audit cannot be done with hand-written accounts for the number of animals in an area.

The transition towards using computers and digital photography, by itself, does not change the underlying amount of work that is needed to manage the catalog. However, as alluded to earlier, automation is needed to make the workload manageable. A primary benefit of using digitized photographic data is that computer vision algorithms can automate laborious tasks. This dissertation introduces the novel concept of *photographic censusing*, a comprehensive and bootstrapable process that uses 1) digital images of animals as input, 2) computer vision algorithms to automate the vast majority of the work needed from humans, and 3) a database to record unique individuals and their respective sightings. Photographic censusing is designed to be scalable; a large geographic area can be surveyed by adding additional, independent photographers, and the method has been experimentally validated *in situ* for large animal populations with thousands of members. Table 1.1 provides a summary of photographic censusing and a comparison to existing methods.

1.1 Animal Detection

Two high-level computer vision components, detection and identification (ID), are required to perform an automated census of animals from photographs. Most of the attention is paid to state-of-the-art identification algorithms in animal censusing, but the role of detection is vital as a required pre-condition for ID, and it is considered carefully here. The detection component analyzes the original photographs and produces a collection of smaller, more focused cropped regions – called *annotations* – around the animals that are of interest for the census. The identification process then takes the annotations and groups them into a database of unique individuals. The detection component must be able to do the following tasks automatically:

1. locate an indeterminate number of animals in a photograph,
2. determine if the animal is relevant to the census (e.g., the desired species),
3. remove visual information that may distract or otherwise confuse identification, and
4. filter out animals that are ultimately not identifiable (not useful in a census).

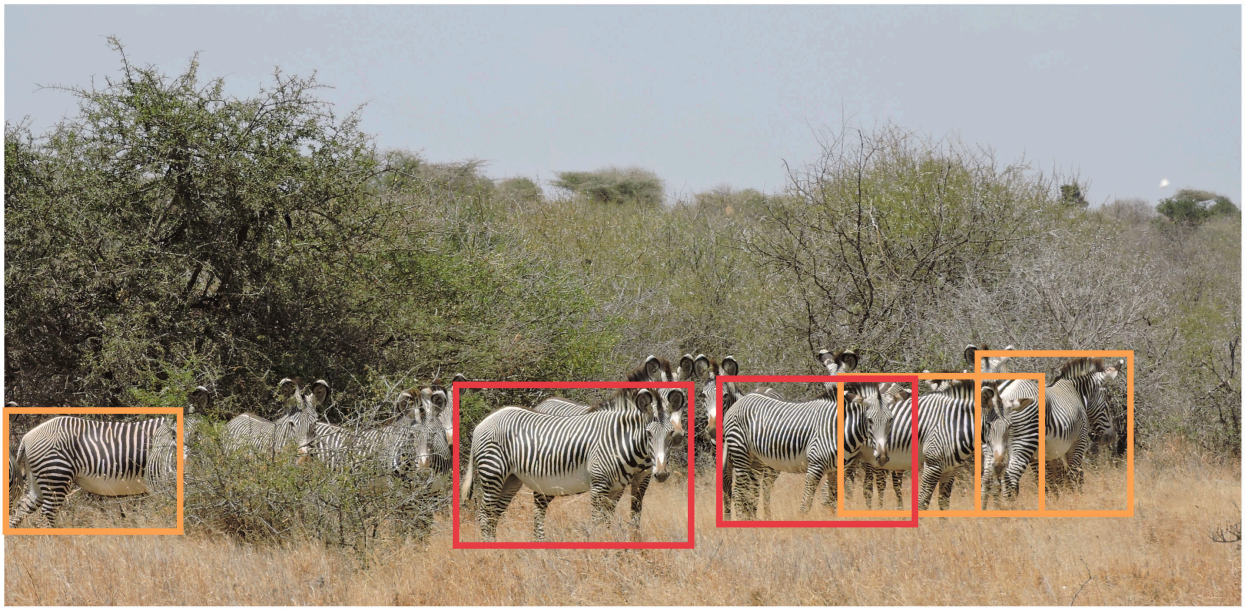


Figure 1.2: An image of a herd of Grévy's zebra in Kenya. The computer vision task of detection is very challenging when considering overlapping animals, each with a different pose and level of occlusion. The red boxes are identifiable animals whereas the orange boxes have a hip or shoulder region obscured (both of which are required for reliable and automated ID). All other animals are too occluded or truncated to be identified.

To achieve these requirements, we need to expand the classic definition of object detection in computer vision. Object detection typically refers to the task of placing bounding boxes around *all* objects of interest in a photograph. This formulation places too much importance on completeness. For example, consider Figure 1.2 which shows a dense herd of 15 Grévy's zebras[3]. What is to be considered the "correct" output of an object detection algorithm for this image? The traditional answer would be to produce a complete set of bounding boxes for each of the 15 animals regardless of their size and clarity. However, this would be inappropriate for use in a census because not all animals are identifiable. For example, the obscured animals behind the bush on the left side of the frame are not seen clearly and should not be provided to an appearance-based identification process.

A more suitable definition of object detection would be aware that some annotations are not worth trying to find, recognizing that it is safe to ignore animals that are ultimately not going to be identifiable. For example, in Figure 1.2 the two most foreground animals (bordered in red) are reliably identifiable, with three additional annotations (orange) that may be identifiable in some

[3]The reader is encouraged to try and count the number of animals, including the sliver of an animal on the far left.

cases. All other animals and detections in this image should be considered distractions. A detection process optimized for use in a photographic census should only produce the 2-5 highlighted boxes as output. The detection process also needs to recognize the species, and even the viewpoint, of interest for a census. Just as people's left and right thumbprints are different, the visual appearances on an animal's left and right sides are different. It would be imprudent for an identification process to compare one person's left thumbprint against another person's right thumbprint because they are fundamentally incomparable. Likewise, how could you tell if a right viewpoint zebra and a left viewpoint zebra were the same animal? Without being able to view the same areas on the bodies and compare corresponding stripe patterns, you would likely be unable to decide *yes* or *no*. The detector needs to go beyond simply creating annotations and should actively try to prevent incomparable matches. The detection process is expected to provide a semantic understanding of the annotations given to ID and filter them appropriately to reduce errors and prevent the need for a human to intervene.

An appearance-based identification process can also be fairly sensitive to poorly-formed annotations. One example is a "photobomb", as seen in Figure 1.3. A visual matching algorithm called HotSpotter [13] (discussed later in this dissertation) was used to search a database of annotations for likely matches, and this pair was returned with a high score. It is clear that the identification process is correctly matching the appearance of the same animal between these two annotations, but not between the *intended* (or primary) animals those annotations are meant to represent. The error that identification has made is understandable – and from its perspective of finding visual correspondences between two annotations is perfectly valid – but ultimately incorrect. The error is possible because the ID algorithm lacks a deeper semantic understanding of which areas (and animals) are being matched. In contrast, we can also frame this example as a failure of detection to properly limit the visual appearance to only the most relevant and comparable regions. If the bottom image were cropped to exclude the animal to the right and focus on the body region, this visual match would not have been made. Reducing these types of "incidental matches" improves the overall automation of the identification process because when failures like this happen, it is often left to a human reviewer to find and fix the issue.

In summary, animal detection is tasked with providing a semantic but filtered understanding of the world to ID. There is a trade-off between producing relevant, identifiable, and comparable annotations and the amount of work needed to find and fix mistakes poor annotations cause during identification. Accordingly, filtering out irrelevant, unidentifiable, or incomparable annotations is

Figure 1.3: An example of a Grévy's zebra photobomb. A photobomb occurs when the same animal is matched between two annotations but the *primary* animal in both annotations is different.

similar to pretending the sighting of that animal never happened in the first place. Since photographic censusing, as a sight-resight study, is based on sampling, it does not expect every animal in the population to be photographed. We can thus rely on filtering to control the completeness of the analysis and balance it against the amount of work humans need to do to produce the population estimate.

1.2 Contributions

This dissertation presents an end-to-end process for animal population monitoring at scale. High degrees of automation, and bootstrapable machine learning components, allow photographic censusing to be performed quickly and accurately, enabling a new and realistic option for data-driven animal conservation. This dissertation offers the following contributions:

1. **Animal Detection Pipeline** - a comprehensive detection pipeline for animals for use in photographic censusing. The pipeline is designed to be easily bootstrapable for new species with relatively minimal amounts of ground truth annotations. The pipeline is constructed out of modularized components, including a whole-image classifier, an annotation and part bounding box localizer, a bounding box orientation regression network, an annotation species and viewpoint labeler, an annotation-part assigner, a coarse background segmentation network, and an Annotation of Interest (AoI) classifier. The pipeline is not limited to Grévy's zebra or herding species and can even be used with multiple species of interest in the same image, on overhead imagery, and with camera trap data. The discussion throughout will use Grévy's zebra as a motivating example due to the availability of large-scale and novel ground-truth detection and ID datasets.

2. **Animal Datasets** - five new public datasets for animal detection and ID research. Common public datasets for computer vision tasks like object detection generally do not provide associated ID information when they include boxes of animals. Likewise, animal ID datasets often only include pre-cropped images of animals and rarely focus on herding species. The largest contributed dataset focuses on Grévy's zebra IDs and is highly curated. The dataset aggregates 5,464 real-world images taken from two large censusing rallies, includes hand-drawn and labeled annotations for Grévy's zebra and 22 other species, and provides ground-truth ID labels for 554 unique animals. Two detection datasets are also made available that have ground-truth bounding boxes and other metadata for multiple species. Lastly, the GGR-16 and GGR-18 datasets will also be made available for ecological research.

3. **Census Annotation** - a novel concept that is designed to reduce incomparable and incidental matching during animal identification. The concept is implemented with two components: 1) a Census Annotation (CA) classifier and 2) a Census Annotation Region (CA-R) regression network. The CA classifier filters out unidentifiable annotations and allows ID only to see the most identifiable annotations during a photographic census. The CA-R network creates more focused regions within existing detected annotations, drastically reducing the amount of human effort by increasing the separability of automated ID verifiers.

4. **Photographic Censusing Rallies** - an organized data collection event where "citizen scientist" volunteer photographers are trained and tasked to take photos of animals for two back-to-back days. The results of the Great Grévy's Rally 2016 (GGR-16) and Great Grévy's Rally 2018

(GGR-18) censusing rallies are significant contributions of this work. Those two rallies are a refinement and extension of the proposed methodology used during the Great Zebra & Giraffe Count (GZGC), the focus of the author's master's thesis [2]. The GGR-16 and GGR-18 rally procedures were significantly improved by increasing the automation of the detection and identification processing, streamlining data collection with GPS-enabled cameras, and proving that the original methodology scales to thousands of animals. In addition, the GGR events collected nearly an order of magnitude more images with twice as many contributors than the GZGC. As a result, they offer the ideal database and framework for analyzing the impact of the automated detection pipeline and CA on real-world data.

Lastly, here is a brief overview of the impact of the methods introduced in this work. The original processing of the GGR-16 and GGR-18 census results was completed with large amounts of human effort. The analysis of GGR-18 utilized 10,044 hand-picked annotations, formed a database of 1,972 unique individuals, and estimated that the population of Grévy's zebra in Kenya was 2,812$\pm$171 animals (CI 95%). The processing was done with relatively experimental algorithms at the time and took approximately three months to complete. It involved dozens of volunteers in drawing and labeling annotation bounding boxes, required 18,556 human decisions of annotation pairs suggested by an ID algorithm, and was estimated internally to cost at least $50,000 USD in time and contracted labor. Using the latest detection methods described in this dissertation, together with a new ID ranking algorithm, all of the original 56,588 images were re-processed. Without any human effort, 11,916 annotations were found that showed identifiable, comparable, right-side Grévy's zebra, with the detection processing taking approximately half a day to complete. After approximately 12 more hours of completely automated ID ranking and automated pair review, a total of 1,297 human decisions were requested before the population estimate converged (it took one reviewer approximately 8 hours to complete). The new process estimated 2,820$\pm$167 Grévy's zebra in Kenya in 2018, created a database of 2,022 unique animals (off by +2.5% IDs compared to the GGR-18 database), and took approximately two working days to generate a result with one human reviewer. If we assume that the reported GGR-18 census results were correct, then the new re-processed population estimate was accurate within 0.3% and had a 93% reduction in human effort.

The remaining chapters of this dissertation are organized as follows. Chapter 2 provides a literature review of related work for deep learning in computer vision, supervised detection and classification methodologies, and existing population estimation techniques. Chapter 3 describes

the detection pipeline, its machine learning components and introduces two new datasets for animal detection. Chapter 4 describes the process of photographic censusing, discusses what kinds of problems must be solved when automation is a primary goal during a census, and offers a mathematical framework for estimating a population size when machine learning methods are involved. Chapter 4 also introduces a new evaluation dataset for Grévy's zebra ID that focuses on providing both ideal and compromised (i.e., hard) annotations. Chapter 5 introduces Census Annotations and Census Annotation Regions as a solution to the problem of incidental matching and other challenging scenarios that make human and automated processing more difficult. Chapter 6 applies the concept of photographic censusing in the real world through photographic censusing rallies. The Great Grévy's Rally in 2016 (GGR-16) and the Great Grévy's Rally in 2018 (GGR-18) were two large-scale photographic censusing events that generated population estimates for Grévy's zebra and reticulated giraffe (*Giraffa camelopardalis reticulata*) in Kenya, which are made available as two new ID datasets. Finally, Chapter 7 provides a summary of the presented research, offers a discussion on its role within computer science, and suggests avenues for future work in automated wildlife conservation.

CHAPTER 2
RELATED WORK

This chapter will review the published literature relating to the methods and algorithms presented in this dissertation. This work has three main intersections with previous research: 1) machine learning, deep learning, and neural networks, 2) computer vision applications, datasets, and techniques for image classification, object detection, and segmentation, and 3) animal re-identification for large-scale population monitoring. The work done by the computer vision field is vast, and animal applications represent a small (but growing) segment. In addition, there has been an increased number of papers and interest in cross-applying advanced computer vision algorithms on animals. This interest has grown enough to support new workshops at premier computer vision conferences like ICPR, AAAI, WACV, and CVPR under the general topics of "Computer Vision for Social Good" or simply "Computer Vision for Animals". The research presented in the following chapters fits well into these themes, and, hopefully, the state-of-the-art in automated wildlife monitoring will continue to be pursued and advanced.

2.1 Deep Learning & Image Classification

The domain of computer vision was thrust to the forefront of publicly known computer science applications with the rise of machine learning and, specifically, deep learning and neural networks [14]–[17]. Neural networks excelled at solving classic computer vision problems like image classification [18]–[22], bounding box localization [20], [23]–[25], and object detection [26]–[29] due to their ability to learn complex representations from supervised training data. The work presented here relies heavily on the advancements in neural network design and improvements in training procedures.

One of the tremendous technological advances of the deep learning era in computer vision has been the ability to learn how to represent an image with a feature extractor [30], [31]. Furthermore, the ability to train a neural work end-to-end that can learn an objective (e.g., object classification)

Portions of this chapter previously appeared as: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

Portions of this chapter previously appeared as: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

Portions of this chapter previously appeared as: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

directly from pixels has been a transformative force within the domain. Therefore, it is essential to review a brief history of neural networks and their impact on the computer vision discipline. The following discussion sets the context for the deep learning methods used throughout this dissertation. In addition, it gives a chronological overview of when current machine learning techniques were introduced and why they are still used in modern applications.

2.1.1 AlexNet & Overfeat

AlexNet [18] was the original network that first broke the mold in 2012 of using hand-engineered features for computer vision tasks. The name "AlexNet" is a callback to "LeNet" by LeCun *et al.* [32], [33], which was designed to perform handwritten digit classification [34] for the U.S. Postal Service in the early 2000's. The approach used by AlexNet achieved the lowest error for the classification and localization tasks in the widely popular ILSVRC [35] challenge in 2012. Until that point, the majority of computer vision applications [36]–[38] relied on SIFT [39], Deformable Parts Models [40], and HOG [37] for these tasks. The technique of Krizhevsky *et al.* diverged strongly from the traditional thinking of hand-engineered feature extraction. Instead, AlexNet learned how to create high-dimensional representations from images that optimized a global loss function. The AlexNet network also first employed the use of dropout by Hinton *et al.* [41] in a competition setting to regularize the final model better and prevent over-fitting. Dropout is used to train some of the neural networks in this research.

The basic instruction set is relatively small to compute a neural network layer's forward activations and backpropagation loss derivative. Deep learning algorithms often use hardware acceleration on Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) [42] to drastically speed up the computation needed for training and inference. Since the computation is relatively simple, it can be naturally parallelized across thousands of smaller, less complex compute cores instead of a handful of general-purpose compute cores like what are found in a modern CPU. For example, the use of NVIDIA GPU hardware with CUDA [43] drastically reduces the training time of large neural networks by roughly 1.5 orders of magnitude compared to CPUs [43]. The AlexNet network was so novel and massive for its time that existing accelerator hardware sold on the open market was unable to handle its size. The authors engineered around that problem by training the network on two separate GPUs to avoid hitting a hard memory constraint. The work presented in this dissertation uses NVIDIA GPU hardware and CUDA to accelerate all of the training and forward inference.

Unfortunately, the original AlexNet network definition and training procedure were unpublished when they won the ILSVRC challenge. The authors of Overfeat by Sermanet *et al.* [20] claim a very similar place in computer vision history by replicating this work and being the first to document and publish an implementation of convolutional classification with a multi-layer network. The work of [44] with their ZFNet was mainly based on the AlexNet structure, but with new hyper-parameter tuning techniques, which led them to win the ILSVRC 2013 challenge.

2.1.2 VGG

The runner-up winners of the 2014 ILSVRC were the creators of the VGG network [21], marking a significant improvement in neural network feature extraction over AlexNet and Overfeat. The advantage of the VGG network compared to previous networks was that it was exceedingly deep for its time, at 19 layers compared to the five convolutional layers of its predecessors. In addition, the VGG architecture used smaller 3x3 convolutional layers and 2x2 max-pooling layers throughout the network, simplifying the network's objective significantly and speeding up training time.

2.1.3 Transfer Learning

A significant advantage of the VGG network was that it began the first meaningful exploration of transfer learning [45]–[47] since the authors had difficulty getting the deeper network to converge. The VGG authors first optimized a smaller convolutional network through "pre-training" and transferred the weights to the final network. With its convolutional filters better initialized, the network was then trained in a process called "fine-tuning" to create the final model. The benefits of fine-tuning should not be overlooked: the transferred filters are likely trained for a particular distribution and may apply inefficiently to a new dataset. Updating the convolutional weights of a transferred model with fine-tuning often improves overall performance. Transfer learning has also driven a massive exploration in neural network applications by allowing for convolutional filters trained on a larger dataset to be applied on smaller applications where not enough data exists to train the networks from scratch. We will see the technique of transfer learning applied in the animal detection pipeline and Census Annotation approaches.

2.1.4 GoogLeNet & Inception

The first-place winners of the ILSVRC 2014 challenge was a Google team with their complex GoogLeNet architecture [25], [48]. The network had a 6.67% top-5 error rate, a noticeable improvement compared to the previous year's first-place winning performance of 14.8%. The key insight of the GoogLeNet architecture was the use of "inception modules", which included a collection of multiple 3x3 and 1x1 convolutional filters within a single layer. The use of 1x1 convolutions is a variant of the research by Lin *et al.* with its "Network in Network" convolutions that also had a filter size of 1x1 [49]. The added inception modules effectively allowed the network to be deeper than the VGG network at 22 layers but with significantly fewer convolutional filter weights (roughly 4 million) than the original AlexNet approach (approximately 60 million). It was clear that deeper models generated superior results, but the research and competition communities still had difficulty training deep networks.

2.1.5 Optimization Algorithms

To improve training stability, the GoogLeNet model was trained by replacing the Stochastic Gradient Decent (SGD) optimizer with a different algorithm called RMSProp [50], which was later combined with AdaGrad [51] and published as the ADAM optimizer [52]. Neural network training had relied until then on various versions of Gradient Decent to optimize the initial conditions of the network weights. In general, a neural network model is initialized with a set of randomized weights (ignoring pre-trained weights) for a given initialization scheme [53]–[55]. An input image (or other data source) is given to the network for its feed-forward inference pass, and it outputs a vector of a pre-defined size. A loss function [56] is then used to compute the current error based on the difference from a provided ground-truth label and the network's output. The error loss for the output layer is then used to compute the loss with respect to the penultimate layer's outputs and repeated recursively for all layers in a process called "back-propagation" [57]–[59]. The respective loss for each layer in the network is then used to update the current weights to reduce the overall error, representing one update step.

A more randomized variant of Gradient Descent, aptly called Stochastic Gradient Decent (SGD), was a successful attempt by [60], [61] to speed up training through approximation. Gradient Descent in its purest form has the gradient calculated for the entire dataset and uses a single weight update per epoch. The key insight of SGD is that the network does not need to see the entire dataset to be able to compute a loss gradient that approximates the *ideal* gradient for the current weights.

Seeing a random sub-sample of the entire dataset is sufficient to calculate the loss for a given state of the weights, significantly speeding up the iterative learning process by adding many more update steps. SGD by itself does have a few optimization downsides: it is susceptible to saddle-points [62] and can oscillate wildly in ravines [63], especially when the wrong learning rate schedule is used. To partially combat these effects, a momentum term can be added to the gradient [64], [65] that adds a moving average (typically $\gamma = 0.9$) of past gradients to the current loss derivative. SGD alone without momentum [66] is also theorized not to be able to reliably find good global minima because it can easily get trapped in less optimal local minima. Another consideration with SGD is how large to make the sample size to ensure it is a representative statistical sampling. Mini-batch SGD [67] uses small batches of examples (typically around 128) and averages their loss gradients into a single weight update. There has been extensive evaluation of mini-batch SGD [50], [68]–[70] within deep learning literature, including distributing the iterative training process to parallelize the gradient computation across multiple machines [57], [71], [72].

2.1.6 Regularization

Turning our attention back to the original discussion on image classification and GooLeNet, the authors used the ADAM optimizer because it works well with complex network architectures and is remarkably fast compared to mini-batch SGD with momentum. All of the neural networks presented in this dissertation are optimized using mini-batch SGD with momentum even though it is slower compared to ADAM (see [68]). Other regularization improvements used on GooLeNet such as batch normalization [73] and more aggressive data augmentation [74], [75] schemes allowed the Google team to train such a deep model successfully. The work in this dissertation also applies both concepts for all of the neural network training.

2.1.6.1 Batch Normalization

Batch normalization [73], [76] (also known as "batch norm") plays a critical role in the performance of deep neural network training as it normalizes the output of each layer to have a zero mean and standard deviation unit vector magnitude. In addition, batch norm helps to control run-away activations, oscillations, and exploding gradients [77], lowering training time. When batch normalization is applied to a layer, it learns two additional parameters: γ and β. The γ term is used to scale the activations of a layer, and β is added as an additional, layer-specific bias term. These values are learned from the statistics of each mini-batch. Furthermore, they are expected to

approximate the mean and variance for the entire dataset for a given layer's activations.

2.1.6.2 Weight Decay

The two most common regularizers in neural network training are L1 (Laplacian) and L2 (Gaussian) weight decay. L1 regularization pushes certain weights to be *exactly* zero and is analogous to having weight decay with a Laplacian prior on the W weight matrices:

$$\Omega_{L1}(\theta) = \sum_{k=1}^{L} \sum_{i=1}^{I^{(k-1)}} \sum_{j=1}^{J^{(k)}} \left| W_{i,j}^{(k)} \right| \tag{2.1}$$

$$\nabla_{W^{(k)}(x)} \Omega_{L1}(\theta) = sign(W^{(k)}) \tag{2.2}$$

L2 regularization pushes the weights *towards* zero and is analogous to weight decay with a Gaussian prior on the weight matrices:

$$\begin{aligned} \Omega_{L2}(\theta) &= \sum_{k=1}^{L} \sum_{i=1}^{I^{(k-1)}} \sum_{j=1}^{J^{(k)}} \left(W_{i,j}^{(k)} \right)^2 \\ &= \sum_{k=1}^{L} \left|\left| W^{(k)} \right|\right|_F^2 \end{aligned} \tag{2.3}$$

$$\nabla_{W^{(k)}(x)} \Omega_{L2}(\theta) = 2 * W^{(k)} \tag{2.4}$$

L2 weight decay is used extensively by the research community and used when training the neural networks presented in this dissertation. It is a very effective regularization technique when used with the ReLU [53], [78], [79] non-linear activation function and batch normalization.

2.1.6.3 Data Augmentation

Data augmentation [74], [75] is the process of applying a set of deterministic or randomized operations on an input image before it is to be used as an example when training a neural network. This process can be seen as a method of balancing the signal-noise ratio to help control over-fitting. Standard augmentation operations for image data include: adding exposure and hue changes,

random Gaussian pixel noise, translation, rotation, skewing, horizontal and vertical flipping, color space transformations, and other sources of randomized pixel noise.

2.1.7 Skip-connection Networks

Neural network architectures before GoogLeNet were relatively linear and did not use multiple branches of activations for a given layer. GoogLeNet introduced the very complex (for its time) Inception Module and showed that complex flows of convolutional activations and their error gradients could be calculated and learned. Using this as insight, neural network researchers asked what would happen if a layer was not branched or copied into multiple streams but instead if some layers were skipped.

2.1.7.1 Residual Networks (ResNets)

The last ILSVRC image classification challenge, held in 2015, was won by He *et al.* and their network ResNet (Residual Neural Network) [26]. The authors drastically increased the depth and circuit length of the neural network by using "skip connections" and liberal use of batch normalization throughout the network. As a result, the network achieved a top-5 error rate of 3.57% and was surpassing human-level performance. The introduction of residual skip connections was a breakthrough in the development of neural network model architectures. The chief design challenge at the time was that deeper networks were shown to increase performance, but increasing the depth of the network caused training problems like vanishing gradients and co-adaptation [80]–[82]. The benefit of residual connections is that the network can selectively turn off a convolutional filter by learning the additive identity [83]. The authors showed that the identity is not only easy to learn (especially with L2 regularization), but it also results in more stable and faster training because the skipped convolutional activations become trivial to calculate.

2.1.7.2 Dense Residual Networks (DenseNet)

An extension of residual networks is the work by Huang *et al.* [84] and their DenseNet architecture. The DenseNet model takes the idea of combining activations for a given layer and a skip connection and extends it by combining the activations from multiple previous layers through skip connections. They further show an increase in performance compared to ResNet (at the cost of speed) and argue that the performance increase comes from increased feature reuse and deep supervision learning [81] within the network. The whole-image classifier, annotation labeler, and

Census Annotation models (described in Chapters 3 and 5) use a pre-trained 201-layer DenseNet model as their feature extraction backbone.

The image classification task has essentially been considered solved by researchers, and new work in deep learning since 2015 has focused more on making networks smaller [85]–[87] significantly faster [28], [88], [89], wider [90], or have moved on to more complex tasks like object detection, segmentation, and 3D applications. The foundation mentioned above of robust feature extraction and research in training improvements has led directly to using neural networks for detection tasks.

2.2 Object Detection & Semantic Segmentation

The computer vision community after 2015 pivoted its focus to more complex tasks like object detection and semantic segmentation since improvements on the classification task were diminishing. The task of object detection is defined by the merging of two separate computer vision tasks: bounding box localization and image classification. Object detection is also getting close to being a solved problem, with real-time commodity implementations available on phones [91] and even readily accessible tools for the wildlife conservation community [92]. However, novelty is still being demonstrated for specific use-cases and real-world applications like large-scale animal re-identification. This section provides an overview of relevant methods to the work in this dissertation on animal detection for ID; a comprehensive review of object detection, evaluation primitives, and datasets can be found in [93] and [94].

2.2.1 Detection Before Deep Learning

Before neural networks and deep learning became a ubiquitous solution for object detection, many algorithms employed hand-engineered feature descriptors and classifiers to find objects. This section gives a brief overview of the most common approaches.

2.2.1.1 SVM Classifier on HOG and Sliding Windows

Histogram of Oriented Gradients (HOG) [37] was the pre-deep learning grandparent of feature extraction and object detection [40], [95], [96]. The method applies a fixed-size sliding window across an image and extracts a HOG feature vector for that window. A Support Vector Machine (SVM) [97] is then used to train a classifier and perform binary classification. The windows are applied on a pyramid of multiple resolutions to support multiple scales of object detections [98].

While these detectors could be trained quickly and with minimal data, they also suffered from poor general performance.

2.2.1.2 Deformable Parts Models (DPM)

Deformable Parts Models (DPM) by Felzenszwalb *et al.* [40] is a more sophisticated version of HOG and sliding windows and was widely popular. The DPM algorithm utilizes a 5-point star model (with a unique model per class) that learns a HOG feature classification for the entire image (the root) and latent variables for the locations of five different parts located around the root. The star pattern is designed to "deform" to find parts in slightly different locations and poses in relation to the root for a given object example. After neural networks had become ubiquitous, an attempt was made to merge their feature extraction abilities with DPM. The work of Wan *et al.* [99] and [100] provides an end-to-end trained model for using convolutional neural network features extraction with DPM and non-maximum suppression (NMS) [101]–[103] for object detection. The work of Girshick *et al.* [104] shows that DPM is a restricted version of convolutional neural networks and provides the argument that CNNs are a more capable and expressive formulation of DPM. While implicitly learned parts are not a component of the detection pipeline proposed in this thesis, it does support explicit, manually-defined parts that can be detected as separate annotations and then linked to a body annotation.

2.2.1.3 Hough Random Forests

The use of Hough Forests (i.e., Hough-transform [105] Random Forests) for object detection was demonstrated by Gall *et al.* in [106]. Unlike DPM, the algorithm is somewhat resilient to partial and occluded objects due to its voting scheme [107], [108]. The authors showed that random forests have advantageous training properties and extend naturally to patch-based image textures. They argue that the leaf nodes of a random forest tree can be considered a "discriminative codebook" [109], which are used to generate classification probabilities. Furthermore, by training to optimize for both classification and regression within the same random forest tree, they can learn a spatial relationship of where a classified image patch is likely located in relation to an object's center. The approach is extended by Barinova *et al.* [110] to address occluding objects while others have applied random forests to face, pose, and action recognition [111]–[113]; a comprehensive analysis of Hough Forests is presented in [114]. A customized version of the implementation by Gall *et al.* is evaluated in Chapter 3 as a baseline algorithm against more modern neural network

detection approaches.

2.2.2 Datasets for Animal Detection

Parallel to the rise of advanced machine learning methods was the creation of large computer vision datasets with supervised labels. However, the few approaches that have used neural networks for animal detection have focused on analyzing camera trap photos [115]–[117] and other applications for counting animals [118]–[121]. Exploring animal detection for animal identification often limits the related work to only animal re-identification methodologies, which often lack a detection component or data suitable for training a detector (i.e., pre-cropped images).

The concept of a detection pipeline, while not novel when considering its components separately, has not been comprehensively analyzed or reproduced in other works for animal ID. The detection pipeline is primarily designed to be used with ground-based photographs but can be re-tooled to work with overhead aerial images for the detection of animals [122]–[126].

2.2.2.1 Visual Challenges: PASCAL VOC, ILSVRC & COCO

While the most prominent public datasets do not focus entirely on animals, they often contain bounding boxes for a handful of different animal species or high-level categories. For example, the PASCAL VOC Object Challenge (VOC) [127] was one of the earliest datasets that had thousands of images and bounding boxes for 20 categories, including six animal classes (bird, cat, cow, dog, horse, sheep). The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [35] was foundational for research in deep neural networks as it offered 1.2 Million images for 1,000 object categories, with a non-trivial portion representing animals. The scale and variety allowed the first generation of neural network models to train well and not severely overfit, giving time and diversity for general-purpose convolutional kernels to be learned. Unfortunately, the animal classes in ILSVRC are very general. For example, the synset `n02391049` for "zebra" includes multiple zebra species taken in the wild by professional photographers, zebras seen in zoos, stuffed zebra animal toys, fondant zebras on cakes, and other abstracted forms like "zebra crosswalks". Thus, the utility of this dataset for animal detection with real-world images is limited.

The Microsoft Common Objects in Context (COCO) dataset [128] is a large dataset with 330,000 images for 80 categories, 10 of which are animals. Interestingly, the COCO dataset has instance segmentations for categories like "zebra" and "giraffe", which can train segmentation networks. The detection pipeline is designed to be bootstrapped and evaluated without the need for

segmentation ground-truth, however, because fully annotated segmentation data is very laborious to annotate. Therefore, all of the methods herein are focused on bounding boxes.

2.2.2.2 Camera Traps & Citizen Science

Other than large challenge datasets exists, there are community-based projects like Zooniverse's Snapshot Serengeti [129], [130] that use citizen science [131]–[133] to annotate camera trap data for 40 African species. The iNaturalist [134] project also uses citizen science to gather and label image data for various animal species. These projects offer lots of data but do not support bounding boxes for animals and therefore also do not offer ground-truth animal ID data. One of the primary benefits of using citizen scientists is that a large number of volunteers can be used to survey a large area [135]–[137]. The Labeled Information Library of Alexandria: Biology and Conservation (LILA BC)[4] project run by Microsoft's AI for Earth initiative is a public repository of animal datasets for conservation. The vast majority of the datasets listed in this repository are based on camera trap imagery and are often limited in their use for detection and animal ID. New applications that use camera-trap datasets [130], [138]–[141] for training show that algorithms can successfully classify camera-trap imagery with computer vision and be a foundation for count-based population estimates.

2.2.2.3 Bootstrapping, Active-learning & Instance-based Learning

A good part of the work in this dissertation is concerned with curating animal ID datasets. However, the protocols surrounding the collection of hand-labeled ground-truth bounding boxes share similarities with bootstrapping detection algorithms [75], [142]–[144] that perform weakly-supervised learning [145], [146]. One highlighted example is the Annotation Interface for Data-driven Ecology (AIDE) project [147] that allows for the machine learning models to be quickly trained as annotated data is being generated, similar to instance-based learning algorithms [148]. The proposed method uses whole-image species classifications to train a whole image classifier and limited human interaction to refine proposed bounding box candidates. This technique can be viewed as a relaxation of one-shot [44], [149], [150] and few-shot [151] learning. Most strikingly, the bounding box refinement problem has been addressed by [152] that shows meaningful speedups in human interactions compared to bounding box regression by hand.

[4]LILA BC - `http://lila.science` (Accessed: Oct. 29, 2021).

2.2.3 Two-Stage Detection with Region Proposals

The earliest deep learning approaches in object detection were created to quickly capitalize on the wild success of their respective winning image classification methods [18], [20], [153], [154]. For example, the early winners of the ILSVRC image classification challenge also saw winning detection solutions by densely applying their neural networks with a sliding window across the image. These methods were relatively crude as they did not fundamentally address detection as a separate task but simply as a brute-force reformulation of the image classification task. These types of two-stage detectors became popular, however, as classification accuracy rapidly improved. A two-stage detector uses an algorithm to solve the localization problem first and feeds candidate bounding boxes to a second algorithm for classification (or suppression). We will explore both salient-based bounding box localization algorithms and deep neural networks that can be used to propose regions around objects for use in a two-stage detection process.

2.2.3.1 Deep Saliency & Attention

In computer vision, the concept of saliency (or "visual saliency") [155]–[157] is the idea that particular objects or items in an image draw a significant amount of attention from the eye. For example, attention is generally pulled to subjects in motion, the most prominent object in the frame, or an object that "pops out" with an abnormal appearance [158]. The critical insight is that salient object detection is class agnostic, and an algorithm can be trained to predict a set of *classless* bounding boxes around things of interest. The salient bounding boxes are then given to a second image classification network to construct the final object detections (two-stage detection). Various pre-deep learning methods have been used for salient object detection, including the use of minimum spanning trees [159], edges [160], BInarized Normalized Gradients (BING) by Cheng *et al.* [161] for speedy region proposals, and bottom-up segmentation algorithms like Selective Search by Uijlings *et al.* [162].

Object saliency with deep learning, also known as deep saliency [163]–[168], has shown to be a powerful tool for suggesting candidate bounding boxes for detection. The work by Kümmerer *et al.* [169], [170] began the first steps of exploring deep saliency with their Deep Gaze network, which borrowed the architecture and transferred weights of AlexNet [18] to create a saliency map of the input image. The parallel work by Liu *et al.* [171] on deep hierarchical saliency network (DHSNet) was also among the first to train an end-to-end neural network to produce saliency maps. AttentionNet by Yoo *et al.* [172] worked in a slightly different manner in that

it aggregated many different sources of salient and weak detection outputs to construct its final detection predictions. Work has also been done to combine local and global contextual information for more accurate saliency maps [173]–[176], take advantage of an attention mechanism more directly [172], [177]–[181], support multiple resolutions [182]–[184], and be able to run in real-time applications [185]–[187]. The Annotation of Interest (AoI) classifier, presented as a component of the detection pipeline in Chapter 3, has an architecture that is structurally similar to Overfeat [20] but is trained on an objective that is more closely related to deep saliency and attention networks.

2.2.3.2 R-CNN & Region Proposal Networks (RPN)

Region Proposal Networks (RPNs) [188], [189] are specialized neural networks that separate the classification task from object detection and focus on only the localization of bounding boxes. RPNs share a similar design goal with object saliency; both are trying to propose class-agnostic bounding box locations for objects, but with the distinction that RPNs often share weights with a neural network image classifier. Similar to salient object detectors, the proposed regions are classified using an image classification neural network to form the final detections.

One of the first neural networks to use region proposals was Region-based Convolutions Neural Network (R-CNN) [190]. The Selective Search [162] algorithm was used initially as input to R-CNN as a preceding region proposal algorithm, but it was prohibitively slow and could obviously not share weights. As a result, the architecture of R-CNN was updated to add a dedicated RPN neural component for object localization [24] alongside an updated classification component [26]. The downside of this network was that the two-branch structure (RPN and classifier) meant that it needed to be trained with an alternating procedure, optimizing either the RPN or the classifier at a given time. This design led to training instability but was still a meaningful improvement in speed and accuracy over using an external region proposal algorithm. A third iteration of the R-CNN detector (called Faster R-CNN [29]) uses a combined training procedure and is the winner of several tasks in ILSVRC 2015. The work in this dissertation analyzes Faster R-CNN's two-stage detection performance using an off-the-shelf implementation.

The design of having a separate component within the network to produce bounding box proposals has been explored by other works. For example, the DeepProposal model by Ghodrati *et al.* [191] and Feature Pyramid Networks (FPN) by Lin *et al.* [192] uses the intermediate activations between layers of an image classification network to find potential object candidates at various scales and perform Non-Maximum Suppression (NMS) to produce a final set of boxes for classification.

The refinement of the R-CNN approach also continues by taking better advantage of the image classifier by training the network to work as a cascade of classifiers [193], [194], where earlier layers discard easy negatives and focus on parts while deeper layers can specialize on large objects.

2.2.4 Single-Stage Detection

Single-stage detectors (also known as single-shot detectors) [23] take a step back and examine what the best neural network structure should be for a detector without being dependent on preconceived designs inherited from image classification networks. In contrast with two-stage detectors, single-state detectors predict a single, combined result of bounding boxes and classifications without needing two inference steps or intermediate region proposals.

2.2.4.1 You Only Look Once (YOLO)

One of the first neural network solutions that was able to train a unified region proposal component with object classification is called, humorously, You Only Look Once (YOLO) by Redmon *et al.* [28]. The YOLO network is designed to predict an *N*x*N* grid of cells (typically 7x7) where each cell assigns itself an object classification label and produces *M* bound box predictions. Each bounding box has a 4-tuple regression prediction for the box's location and a salient "objectness" confidence score (similar to [195]). The final predicted bounding boxes are generated by multiplying the classification label scores for each cell by the object confidence scores for each of its bounding boxes. The ability to train YOLO as a unified pipeline makes it advantageous for real-world applications due to its efficiency and lack of additional training infrastructure (no need for alternating between branches during training like R-CNN). Due to YOLO's relatively simple network architecture without an RPN, its authors reported real-time performance using GPUs.

However, YOLO's integration of bounding box predictions into a unified network comes with downsides: a complex loss function, additional hyper-parameters, an unpredictable error gradient at the start of training (which often diverges), and a lack of multi-resolution detections. To address training instability, YOLO uses transfer learning and a process called "burn-in" that starts with a relatively small learning rate to warm up the network before the actual training. YOLOv2 [196] was introduced to address common failures made by the original network; YOLOv2 adds Batch Normalization to increase training stability and lessen the need for burn-in, adds training and inference at multiple scales, and starts using anchor boxes [197]. An *anchor box* is defined as one of *k* centroids when the ground-truth bounding boxes are clustered. The use of anchor boxes allows

the model to focus on regions and sizes of boxes that are likely to be seen instead of attaching them to an arbitrary underlying grid cell. Finally, YOLOv3 [198]) was introduced to modernize the approach of YOLOv2 with a better feature extraction backbone using ResNets and adds support for three separate scales of predictions to localize smaller objects better (similar to [199]). For this research on the detection pipeline, the YOLOv2 model is analyzed against Faster R-CNN for animal detection.

2.2.4.2 Single-Shot Detectors

Shortly after YOLO was published, the Single Shot Multibox Detector (SSD) by [167] was introduced as an alternative single-shot detector. The main difference between SSD and YOLO is that it uses a fully convolutional neural network (FCNN) [27] while still being able to achieve real-time detection performance. The accuracy was also a bit higher than YOLO (version 1), and it rivaled two-stage detection approaches like Faster R-CNN in terms of accuracy while also being substantially faster at inference. The design of SSD, like others [200]–[203], takes advantage of a unified convolutional structure and introduces bounding box prediction at intermediate layers for multi-scale detections. Other approaches use Receptive Field Blocks [204] to enhance feature selection for object detection, and the Trident Network [199] approach learns a three-branch, single-shot neural network that generates small, medium, and large bounding box predictions. More recent single-shot detectors attempt to remove the need for anchor boxes entirely and instead use keypoint triplets [205] or hourglass designs [206]–[208].

2.2.5 Semantic & Instance Segmentation

Novel bounding box proposal and single-shot networks became less frequent around 2018 and 2019 as incremental improvements to object detection performance diminished. The fundamental problem is that bounding boxes are rigid and limiting shapes – detection failures became more nuanced [198] because boxes are sometimes hard to draw and locate consistently. It was clear that to advance the state-of-the-art for object detection, a reformulation of the objective was needed: the community needed better, more precise bounding boxes. It is not so much that existing bounding boxes in large datasets were not labeled correctly, but rather that bounding boxes were too coarse of a concept, and access to more intimate details was needed.

Semantic segmentation is the task of labeling the exact pixels that belong to a given class category. Semantic segmentation has historically been used as a means for object detection [209]–

[213], locating parts [214], and have been implemented using a range of techniques, including Fisher vectors [215], fully connected CRFs [216], and graphs [217]. For example, given a picture of Times Square in New York City, we could ask a person to paint all cars with red paint, buildings with blue paint, sky or water with yellow paint, road and sidewalks with purple paint, people with green paint, and everything else with orange paint. The goal would be to paint every pixel in the image with an assigned color. If we want to segment out each unique car in the image, however, painting all of the cars with a single red color offers insufficient detail to perform the task. Instance segmentation is an enhancement of semantic segmentation where each instance of a given class is also annotated. In our New York example, an instance segmentation would ask a computer to color all cars with different shades of red so that the boundary for all cars is defined down to the pixel.

The required level of detail for segmentation is much more involved and precise than drawing a bounding box for each object, making it much slower to gather. The success of segmentation techniques has been parallel to the creation of large datasets like Microsoft's Common Objects in Context (COCO) dataset [128] that have spent the time to add instance-level segmentations for a large number of images and classes. Likewise, other methods have shown that it is possible to simulate color images and ground-truth segmentation data for training [218]–[220]. While this dissertation does not use semantic or instance segmentation techniques, it is related to the coarse background segmentation component in the detection pipeline. The results reported here suggest that instance segmentation will allow for even more automated photographic censusing methods in the future. However, the resources and funding of conservation groups are often minimal, and it is difficult to realistically expect fully segmented ground-truth to be annotated at large scales for novel species. To maximize the real-world usefulness of the methods presented here, the focus on using annotated bounding boxes (with select metadata) is key to keeping them adaptable for new species and a feasible option for wildlife conservation groups.

2.2.5.1 Fully Convolutional Neural Network (FCNN)

A Fully Convolutional Neural Network (FCNN), introduced by Long *et al.* in [27], is a special type of neural network that has no fully connected dense layers. The benefit of having no dense layers is that the network is not rigidly set to a fixed input or output size. This feature can be exploited by applying the network in a fully convolutional fashion across a larger input image implicitly, and the network does not need to resort to any type of fixed-sized sliding window or shift-and-stitch techniques [20], [153]. The FCNN has similarities to the All Convolutional Network

by Springenberg *et al.* [22] in that the network architecture is comprised entirely of convolutions with no fully connected dense or pooling layers. The design of the FCNN makes it a flexible platform for image classification, region-based object detection [221], and a natural candidate for segmentation [222]. The detection pipeline has a coarse background classifier that is implemented as an FCNN and uses semi-supervised learning [223] on bounding boxes. There is not currently a component in the detection pipeline that relies on having full object segmentations for training data because rectangle bounding boxes are sufficient for all training.

2.2.5.2 U-Net & Mask R-CNN

The work of Ronneberger *et al.* [224] proposed the novel U-Net architecture with its convolution, embedding, and up-scaling layers. U-Net uses a single-shot process to generate semantic segmentations directly from input images. The network shares outputs from the convolutional feature maps to their corresponding up-scaling segmentation maps for the same resolution. The use of up-scaling branches led to further development of de-convolutions [225]–[227] and their use in semantic segmentation. Furthermore, the work by Yu *et al.* [228] on dilated residual networks allowed the network to learn how to effectively up-scale images. As for two-stage segmentation methods, Mask R-CNN [229] extended the author's previous work on R-CNN to produce a semantic segmentation as outputs of the RPN. The Detectron [230] approach uses existing bounding boxes or rough semantic segmentations to create instance segmentations. The approaches of U-Net and Mask R-CNN are very popular (with over 28,000 and 12,000 citations, respectfully) and have been used on animal detection [231]–[233] and aerial counting [118], [234], [235].

2.3 Animal Re-Identification & Population Estimates

Human re-identification (re-ID, also referred to as "biometrics") [236]–[239] has long been the interest of computer vision applications and has natural cross-applications with animal re-identification. While the image classification and object detection techniques we have discussed can find animals and determine their species, it is difficult to apply these concepts directly to identifying unique individuals. New algorithms are therefore needed to solve animal identification as a dedicated task. For example, a detection process is still needed to filter relevant images and sightings of animals. The job of an identification procedure is to build a searchable database of repeat sightings of the same animal and calculate a population estimate.

Historically, population estimates have been done entirely by hand, using counting-based

methods [129], [130], [139], [240], physical tags or collars [241]–[245], or manual description codes [246]–[248]. These estimates are typically custom, one-off efforts and do not have uniform collection protocols or data analysis. Because datasets are often curated by hand, they tend to be focused on a small number of individuals [249] or focus on animals with few repeat sightings [250]. One of the most challenging barriers with performing population estimates with deep learning is that there is a structural mismatch in target species between large datasets for animal re-ID (that show *pre-cropped* images for at least hundreds of individuals with repeat sightings of each animal over time) [251]–[253] and public datasets for animal detection (with at least thousands of annotations and original images that are seen in different locations, but without ID) [130], [254]–[256]. Attempting to build deep learning algorithms for a single species can be severely limited by not having access to large-scale datasets for both the detection and identification tasks. As presented in this dissertation, the concept of photographic censusing is a bootstrapable and end-to-end framework for generating ground-truth animal detection datasets with curated animal IDs.

While this dissertation does not contribute new animal identification methodologies, it does use them in its photographic censusing process. A brief overview of animal identification is given below, but the reader is encouraged to explore a more comprehensive history provided by Ravoor *et al.* [257], Hoeim *et al.* [258], and Weinstein [259].

2.3.1 Animal ID Ranking & Verification

Animal re-identification (also known as "animal re-ID") [260] can be broken up into two tasks: ranking and verification. Identification ranking [261]–[264] is the process of querying the image of an animal against an existing search database of previous encounters to find visual-based matches. The most confident matches are returned in rank order, with the highest-scoring database example in position one (i.e., rank-1). Identification verification [265]–[270] is quite different as there is no need for searching: verification asks if two presented animals are the same or not, regardless of why the pair is being compared or how it was found. For example, if you were given a grainy photo of a person's face and a pile of 100 driver licenses, you could rank the licenses according to the people you felt were the closest to matching the reference image. Maybe you would first partition them by gender, then sort by age, then organize by skin color, etc. and then narrow the candidates to the handful you felt were the most likely. Likewise, you could also be given the same grainy face photo and one license and asked to make a *yes* or *no* decision on if those two photos represent the same

person. We can realistically expect an ID verification algorithm to be much faster than ID ranking; ranking images with a verifier through brute-force is possible but can quickly become infeasible as the database grows. In other words, both tools are useful for human and animal ID as they can optimize for two very different goals. If both of these tasks work relatively well for a given animal species, it is possible to build automated systems that can generate a population estimate, as this research will demonstrate.

The challenge for identification ranking is that not all species show the same kinds of visual information for matching, even though texture-based matching is successful across a variety of species [271]–[279]. For example, a zebra has high-contrast stripe textures visible across the body that do not change over the life of the animal, a perfect example of a species that can be matched with visual ID [261], [280]–[283]. On the other hand, a green sea turtle has lots of texture on its shell, but those patterns change slowly over time (like rings of a tree). The overall color and appearance of a sea turtle shell can also change based on the animal's diet. The face and flippers of a sea turtle, however, are covered with small patches (called "scutes") that are reliable for pattern-based ID [284], [285]. It is important to recognize that not all parts (like a shell) of an animal are reliably useful for ID over time. Some species may require more specialized attention by a detector to find specific parts of the animal.

Like a bottle-nose dolphin or an African elephant, some animals do not have stripes, spots, or intricate patterns for pattern-based identification. The lack of texture, however, does not necessarily make these species unidentifiable. Rather, it asks if different paradigms of ID algorithms can make ID work for those species. Animal ID algorithms can be designed to focus on identifiable features like the outline of a dorsal fin [286], the jagged nicks and notches of a whale fluke [287]–[290], or a fanned-out ear of an elephant [291], [292]. Animals that do not have intricate patterns or detailed contours (i.e., local features) may still offer large structures or definition-less blob patterns (i.e., global features) that can be used for ID. For example, the bonnet callosity pattern of right whales [250], [293]–[296], or the Rorschach-like underbellies of giant manta rays [263], or the unique constellations of whale shark spots [297]–[299] can be used for recognizing and distinguishing individual animals.

It also seems evident that some species, like the American red squirrel (*Tamiasciurus hudsonicus* or Grant's gazelle (*Nanger granti*) in Africa, are simply beyond the practical ability of visual ID to recognize individuals. We should recognize that the abilities of any visual ID algorithm are fundamentally tied to a human's ability to confidently decide if two sightings show the same animal

or not. If a human was presented with two images of squirrels, it seems improbable that a reliable "same" or "different" decision could be made without the aid of scarring or a deformity. This begs the question, "*how could a ranking algorithm's results, even from a perfect oracle, be trusted if a human was unable to tell if the rank-1 match was correct or not?*" While we can consider ID ranking to be a *super-human* task – something that is expected to surpass human-level performance – a human's ability to verify pairs should be a bellwether for ID feasibility. If humans cannot accurately verify pairs of annotations for a species, then that species is categorically incompatible with visual ID methods and is a better candidate for a more invasive or abundance-based ID alternative. The methods described here consider unidentifiable animal species outside of the problem scope for visual population monitoring and photographic censusing.

Other than body texture and edge contours, other approaches have treated animal ID in a similar way to human face ID [300]. Animal faces have been shown to be trackable in video frames [301], [302] and moderate success has been shown when applying modified face ID algorithms to chimpanzees [303]–[305]. The biggest issue with chimp face ID is that the populations are fairly small, and the broader impact on other species is not very well understood. Apes are not the only candidate for face ID; lemurs [306] have also worked with face ID methods and the whisker patterns of brown bears [307], polar bears with HAAR-features [308], [309], and lions [310] have shown success for identification.

The various methods for animal ID are not a direct focus of this dissertation, but some baseline algorithms are needed to demonstrate the impact and success of the contributed methods. Some algorithms, like triplet-loss networks [311]–[313], require significant amounts of training data and need to be bootstrapped by algorithms that do not rely on deep learning. The following algorithms were co-developed with the detection pipeline and photographic censusing methodology presented in this work and are selected as representatives for detailed analysis:

1. HotSpotter [261] - a texture-based ranking algorithm that uses local features on areas with sharp changes in contrast. This algorithm uses SIFT features [39] at its foundation and does not need to be trained with a deep-learning algorithm, meaning it can be run on new species out-of-the-box with minimal tuning.

2. CurvRank [314] - a curvature-based ranking algorithm that matches local segments of an edge contour. This algorithm requires training data to predict outline contours but does not rely on comprehensive ID data for training. While this algorithm cannot be run on new species

completely natively, it can cross-apply its pre-trained models on similar features (i.e., dorsal fins look very similar, regardless of species).

3. Verification Algorithm for Match Probabilities (VAMP) [13] - a random forest verification algorithm that uses hand-engineered features for comparing two sightings. This algorithm does require training data for ID comparisons but can be trained from a small (and converged) database of animals due to its data mining procedure.

4. Pose-Invariant Embeddings (PIE) [263] - a triplet-loss algorithm that creates a global embedding feature for distance-based ranking *and* verification. This algorithm is often the most accurate for a given species but requires extensive training data to train. New species cannot be ranked (or verified) by PIE until an algorithm like HotSpotter or CurvRank builds a preliminary dataset of IDs first that can be used to train the feature extraction and embedding.

The proposed components and methods in this dissertation are designed to be modular and general-purpose and may be used with other ranking or verification algorithms. An overview of these four algorithms (and their related work) is offered below.

2.3.1.1 HotSpotter & VAMP

The work by Crall [13] performs texture-based animal ID ranking by comparing SIFT descriptors [39] that are extracted at keypoint locations [315] for an annotation. Foreground-background segmentations from the detection pipeline (see Section 3.5) are used to weight these extracted keypoints, and the resulting descriptors are gathered into an approximate nearest-neighbor (ANN) search data structure [316]. A new annotation can then be queried against the ANN index to find descriptors similar to others in the database. Matches in the sparser regions of descriptor space (i.e., those that are most distinctive) are assigned higher scores using a "Local Naive Bayes Nearest Neighbor" method [317]. The scores from the query that match the same individual are accumulated to produce a single score for each animal. A post-processing step then spatially verifies the matches and re-ranks the returned list of individuals [318] (as will be defined and discussed in Chapter 5).

In addition to the HotSpotter ranking algorithms, the Verification Algorithm for Match Probabilities (VAMP) verification algorithm was also developed by Crall [13]. VAMP is trained as a random forest classifier [319], [320] on a hand-engineered feature vector and produces a decision of "same animal", "different animals", or "cannot tell" for a pair of annotations. The model is swift and is relatively accurate for well-formed annotations.

2.3.1.2 CurvRank

The CurvRank algorithm by Weideman [262], [291], [314] uses a U-Net [224] architecture to extract a coarse contour and a self-supervised [321] CNN to refine that edge into a fine contour. The contour is then converted into a series of descriptors with a novel, digital curvature-based feature extractor. The descriptors are placed into a nearest neighbors search structure and matches can be queried. A similar algorithm called FinFindR [287], [322] also works on extracted contours and uses A* [323] to produce a trailing-edge segment for dorsal fins. The algorithm then uses a pre-trained classifier to recognize a unique fixed set of individuals, requiring substantial training data and a need to retrain periodically.

2.3.1.3 Pose-Invariant Embeddings (PIE) & Triplet-Loss Networks

One of the most recent techniques to perform animal ID ranking is a triplet-loss network [311]–[313]. A triplet-loss network aims to learn how to represent an animal's identity directly and extract a feature embedding that can be compared with other embeddings (without the need for normalization). This design has seen success in animal classification by normalizing the pose of birds [324] and was cross-applied to instance recognition (i.e., re-identification) for animals [249], [325], [326]. In contrast to HotSpotter or CurvRank, the intermediate features and descriptors cannot be visualized, but the distance between two features does not need to be normalized before clustering. The ability of triplet-loss networks to learn a global feature embedding makes it generally more accurate and faster than methods that use hand-engineered features. However, it comes at the cost of needing large amounts of training data.

Triplet-loss networks are an enhancement of Siamese networks [327]–[329] and are trained by mining a triplet pair consisting of a reference image, a positive example, and a negative image. During training, the network is tasked with learning how to do feature extraction for embeddings; ideally, the distance between the reference and the positive embeddings should be small, while the distance for the reference and negative pair should be large. The Pose-Invariant Embeddings (PIE) algorithm [263] has an additional component that allows multiple poses for the same animal (left and right) to be learned within the same model. This dissertation uses HotSpotter and PIE to rank annotations of Grévy's zebra and build a curated database. The VAMP algorithm is also compared against PIE as a verifier in an analysis of how much work can be automated during a population census.

2.3.2 Animal Population Estimates

The field of animal population estimation is much older than the era of deep learning, stretching back to 1896 and the work of Johannes Petersen and his mark-recapture ecological studies [330] on European plaice (*Pleuronectes platessa*). Since that time, various statistical techniques have been used for sampling animal populations and estimating error. The detection pipeline and other methods are designed to be used as black-box components within a larger censusing framework. Various frameworks [2], [139], [331] in the conservation literature have included computer vision components as well.

2.3.2.1 Capture-Mark-Recapture

Mark-recapture is used to estimate the size of an animal population [10], [11], [332]–[334]. Typically, a portion of the population is captured at one point in time, and the individuals are marked as a group. Later, a second population capture is performed, and the number of previously marked individuals is counted and recorded. Since the number of marked individuals in the second sample should be proportional to the number of marked individuals in the entire population (assuming consistent sampling processes and controlled collection biases), the size of the entire population can be estimated. [281] The population size is estimated as the ratio of marked individuals during the first and second captures against the number of resighted individuals. Thus, the formula for the simple Lincoln-Petersen estimator [335] is:

$$N_{\text{est}} = \frac{K * n}{k} \tag{2.5}$$

where N_{est} is the population size estimate, n is the number of individuals in the first capture, K is the number of individuals from the second capture, and k is the number of *recaptured* individuals that were marked from the first capture. There also exist more sophisticated extensions to the formula that account for various known sources of error [12], [332], [336].

Applying the Lincoln-Petersen estimator requires that several assumptions be met. The estimator expects that no meaningful births, deaths, immigrations, or emigrations have taken place. Further, the sightability of individuals must be equal between photographs. Sampling back-to-back days reduces the likelihood of violating the first two assumptions for most large mammal species. For photographic censusing, we can assign multiple teams of volunteers to traverse the same survey

area to attempt to increase the overall number of sightings. More sightings on the first day mean better population coverage and increased resightings on the second day give a more confident population size estimate. By intensively sampling a survey area with many photographers (that may haphazardly overlap), the expectation for equal sightability is high and identical for any given individual in the population. Therefore, all of the required assumptions for the Lincoln-Petersen estimator can be satisfied for a photographic census. A two-day collection is structured into a public "rally" that focuses specifically on upholding these sampling assumptions and coordinating the help of volunteers.

This work explores a passive variant of mark-recapture that is based entirely on photographs called sight-resight [337], [338]. The entire photographic censusing technique can be viewed as an automated and large-scale implementation of a sight-resight study. By tracking individuals, related to [339], [340], the proposed method can make more confident claims about the population. The more individuals that are sighted *and* resighted, the more confident the population estimate and robust the ecological analyses will be.

2.3.2.2 Graph ID & Local Clusters and Their Alternatives (LCA)

We now must consider how to associate and curate annotations into their respective IDs accurately. The immediate question is, "*how do we use animal ID ranking and verification algorithms as tools to build a database of animal IDs?*" One naïve solution is to begin with an empty database and build it incrementally by adding one annotation at a time. Each time a new annotation is added, it is expected to be matched against the current database. The ranked ID results for the new query annotation can be passed to a verification algorithm to 1) automatically decide which database annotations (by pairing them up with the original query annotation) show the same animal or 2) filter and reorder the ranked results for human review. At any point, a human reviewer could also be presented with the same pair of query and database annotations as the verification algorithm to get a ground-truth decision. This design allows for human-in-the-loop [341]–[344] verification of the database as it grows, and human reviewers can be used to help correct for any errors made by the underlying machine learning algorithms [345]–[347]. If a confident match is found, it is added to an existing ID in the database. Otherwise, if no match is found, then a new ID is added to the database. This process is termed *one-vs-many agglomerate matching* and is one of the easiest to implement for large animal databases [281].

This process, however, does not have any built-in way to identify and correct ground-truth

errors in the database. Database errors can be introduced and may accumulate over time if the ranking algorithm fails to retrieve a correct match from the database where one exists (false negative). An error may also be introduced when a verifier automatically decides that annotations for two different animals are the same individual (false positive). A human reviewer can also make mistakes and, for example, could decide that two annotations of the same individual are different animals (erroneously increasing the total population size by 1). As the database grows, ID mistakes can become non-trivial in size and sometimes require substantial amounts of effort to fix. One example of such a database mistake is a "snowball". This type of error can be expected for herding species where annotations overlap and is created when two actual individuals are incorrectly matched together under the same ID label. The error, in turn, makes it more likely for a third individual to be matched as the same name in the database, and so on until many individuals are represented by one name label (decreasing the population size). Fixing this type of error is laborious because it requires the one big name to be split into an unknown number of smaller names for each distinct individual. When we constrain ID matching to only an agglomerate process – always making new animal IDs or adding to existing animal IDs – it becomes exceedingly difficult to know if (or indeed how many) errors there are in the underlying database over time.

The end goal of photographic censusing is to create a consistent database of individuals and their respective sightings. This database can be used to estimate the number of animals in the overall population, which can be sensitive to systematic ground-truth errors in the ground-truth ID database. Leaving these errors unaccounted for and unresolved may end up skewing the direction or urgency of conservation action, so it must be addressed. What is needed is an overarching management algorithm that can continually curate an existing database and use *many-vs-many reinforcement matching* to run consistency checks on its current IDs. This database consistency problem is important enough for accurate population monitoring that it demands a dedicated solution, and two algorithms are analyzed by this dissertation: Graph ID [13] and LCA. These algorithms are responsible for ensuring that the current state of the database is trustworthy by enforcing a level of self-consistency. As database errors are found and fixed, the management algorithm should also decide which pairwise verification decisions to send to a human and control how much automation there is during the curation of the database. This type of review is similar to active-learning [348]–[351] since the updated ground-truth IDs can be used to iteratively re-train the underlying machine learning algorithms [352] and improve the overall estimate. The process of continual curation also shares similarities with database visualization for consistency checking [353] and ground-truth data

debugging [354], [355].

The first algorithm, Graph ID [13], allows for the state of a population of animals to be constructed as a graph of annotations (nodes) and pairwise decisions (edges). The nodes of the graph are all expected to be annotations that can be visually matched using a ranking algorithm. Decisions with three possible states represent the edges between two nodes: "same animal", "different animals", or "cannot tell". The goal of the Graph ID algorithm is to construct a consistent graph of positively connected components (PCCs) where there are only negative edges between all PCCs. The algorithm relies on a positive-redundancy measure within all PCCs and negative-redundancy between all matching PCCs to ensure that the database is in a consistent state. This need for explicit redundancy and the possibility of an incomparable ("cannot tell") decision means that the algorithm stops all automated processing when an inconsistency is found, expecting a human reviewer to find and fix the issue. If the verification algorithm is not confident enough to decide a given pair, it is also given to a human for review. Likewise, if a PCC is inconsistent, all of its previously reviewed annotation pairs are given to humans for review until the error is found and resolved. Likewise, since the algorithm requires all (matched) PCCs to satisfy negative redundancy, there is a quadratic increase in the number of negative edges that need to be reviewed by humans. While redundancy is conceptually easy to understand, the Graph ID algorithm places an outsized focus on enforcing it and does not take full advantage of the automated verification algorithm.

The Local Clusters and their Alternatives (LCA)[5] algorithm is developed as an alternative to the Graph ID algorithm and makes better use of the automated verifier. The (experimental and yet-to-be-published) algorithm accomplishes this goal by shifting away from the concept of positive and negative connectivity. Instead, it attempts to measure a cluster's relative stability in comparison to alternative clusterings. In addition, LCA chooses to delay human decision-making for as long as possible. Further, it does not require consistency at all times (and forcing human decisions when a mistake is found). It instead relies as much as possible on automated decision-making to infer what the most likely resolution is. LCA will run a series of trials by splitting the cluster apart and measuring the coherence of a handful of alternatives, and only ask for a human decision when all of the various alternatives are too unstable. In practice, this drastically reduces the amount of human effort to curate a population graph and is a much more efficient algorithm for automated population censusing. While LCA is not a contribution of this dissertation, the work discusses how LCA behaves differently than the Graph ID algorithm and analyzes its failure modes. A large-scale

[5] `https://github.com/WildMeOrg/wbia-plugin-lca` (Accessed: Oct. 29, 2021).

experimental analysis of the LCA algorithm to verify ID datasets is a contribution of this work, as it presents an initial benchmark for the algorithm's performance compared to Graph ID.

2.3.2.3 The Great Zebra & Giraffe Count (GZGC) of 2015

The formalized concept of a photographic censusing rally is a significant contribution of this work. A censusing rally is designed as a two-day event that focuses on collecting many images for a target species and attempts to survey its known geographic area. Citizen scientists [131]–[133] are used as volunteer photographers to increase the overall coverage of the surveyed area, distribute the workload, and overall make data collection more feasible. The image data collected by all participants are then analyzed by machine learning to produce a database of resident animals and estimate the size of the population.

One of the first real-world demonstrations of photographic censusing was The Great Zebra & Giraffe Count (GZGC) of 2015 and is the focus of the author's master's thesis [2]. The GZGC censusing rally was a small case study performed within the Nairobi National Park in Nairobi, Kenya to estimate the local population size of plains zebra (*Equus quagga*) and Masai giraffe (*Giraffa tippelskirchi*). The primary goal of the GZGC was to prove the effectiveness of the general censusing procedure with quickly-trained volunteers and to test the workflow of using automated detection and ID algorithms for real-time feedback to participants. The insights and lessons learned from that event were applied during the Great Grévy's Rally (GGR) [356], [357] to estimate the total number of Grévy's zebra in Kenya. The details and analysis of the GGR photographic censusing rallies in 2016 (GGR-16) and 2018 (GGR-18) are the focus of Chapter 6. To provide a quick summary: the two Great Grévy's Rally events combined collected over 90,000 images and used over 350 participants, compared to around 9,000 images and 50 contributors during the GZGC.

2.4 Summary

The techniques proposed in this dissertation span the disciplines of computer science, computer vision, and ecology and are heavily motivated by the application of real-world population monitoring. The related machine learning work in image classification, object detection, and other semantic computer vision algorithms allows the automated processing of large volumes of images for photographic censusing. Separating the work responsibility into two stages – a detection pipeline followed by a separate identification process – is helpful since it allows for modularized development and dedicated attention when creating machine learning datasets.

CHAPTER 3
ANIMAL DETECTION PIPELINE

This chapter presents a pipeline of modular machine learning components that detect, classify, and otherwise prepare images of animals for use in a visual identification (ID) procedure. The computer vision task of object detection includes the inherent step of finding animals in images but, when used as a prerequisite for animal ID, it needs to be able to do much more than just that. For example, animal detection needs to determine an animal's species and viewpoint so that automated tools consider only annotations that can actually be compared; having ecological metadata like species and viewpoint increases the accuracy and speed of ID by filtering out annotations that could only function as potential confusers. An annotation may also need to be rotated to allow accurate matching or have its background segmented out because it is distracting for algorithms or humans. What is needed is a comprehensive pipeline that can perform a variety of different "animal detection" tasks so that an automated ID process can focus on its tasks of describing, retrieving, ranking, and verifying potential matches of animals.

The animal detection problem can be exceedingly complex: there may be multiple (or no) animals from several different species in an image, some species might not be the target of ID but have a similar visual appearance to the species of interest, some annotations may have poor quality while others may show only parts of the animal, or an animal may be occluded by other animals or vegetation. Furthermore, animals may be seen from various scales, viewpoints, and poses, only some showing identifiable information. The images provided to the detection pipeline may also originate from handheld cameras used by trained ecologists or novices (e.g., tourists, children) with no prior experience taking photos of animals for photographic ID. Images can also be captured by passive collection devices like a camera-trap or an aerial surveying platform. The detection pipeline must account for these challenges and automate the creation of high-quality annotations useful for animal ID or wide-area aerial counts.

The detection problem as applied to photographs of zebras, for example, has several real-world challenges: varying viewpoints, natural and artificial occlusions, overlapping animals (i.e., instinctual herding behavior), non-rigid body structures (legs, necks), and significant changes in

Portions of this chapter previously appeared as: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

Portions of this chapter previously appeared as: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

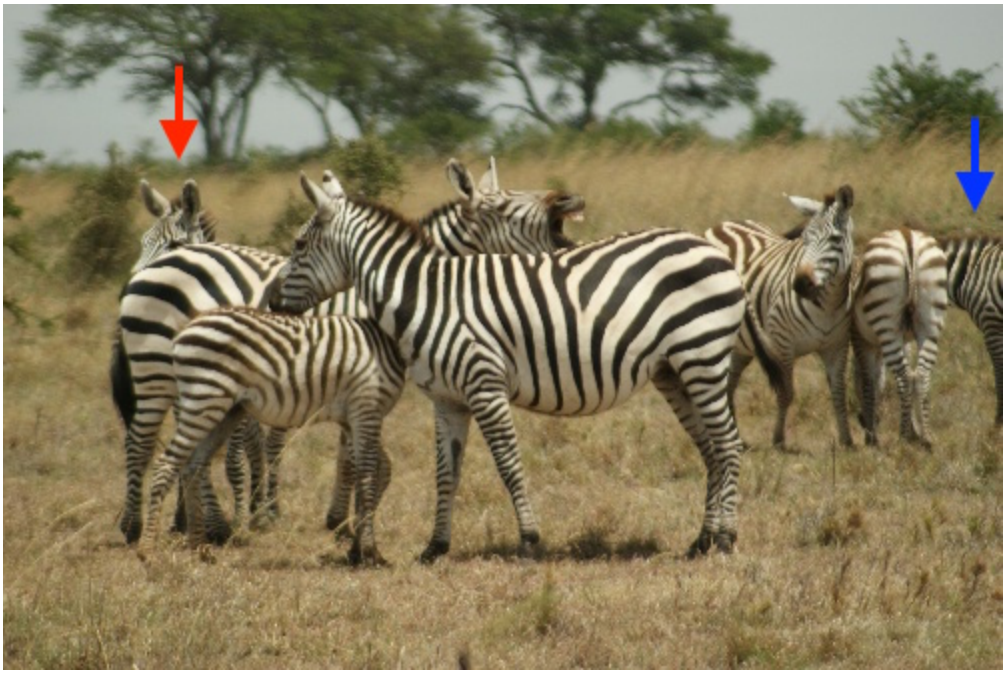


Figure 3.1: The challenges of the detection problem (shown for plains zebras) include varying viewpoints, natural and artificial (image frame) occlusions, and overlapping animals. The image shows 7 individual zebras with 5 differing viewpoints and 5 occlusions of differing severity. The highlighted animals are almost completely occluded, but still clearly discernible. ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

visual appearance (e.g., genetic variations, dust, scarring). Looking at Figure 3.1, the head of the highlighted zebra (red arrow) is visible, but the rest of the animal is almost completely occluded except for one or two legs. The cut-off animal on the far right (blue arrow) only has a small section of neck visible, whereas its neighbor to the left is facing entirely away from the camera. While the challenges listed above are not unique to zebras, they are typical for species that assemble in herds and social groups. Zebras seen in the real world can be frustratingly uncooperative with respect to the task of trying to detect and identify them, which makes them an ideal challenge species for evaluation. Furthermore, these kinds of challenging detection scenarios elevate the problem to a degree of difficulty not often seen in standard computer vision benchmarking competitions like PASCAL VOC [127] and ILSVRC [35].

In response to the challenges outlined above, a five-component detection pipeline is proposed and analyzed in this chapter (see Figure 3.2 on the next page). These components are, in order of

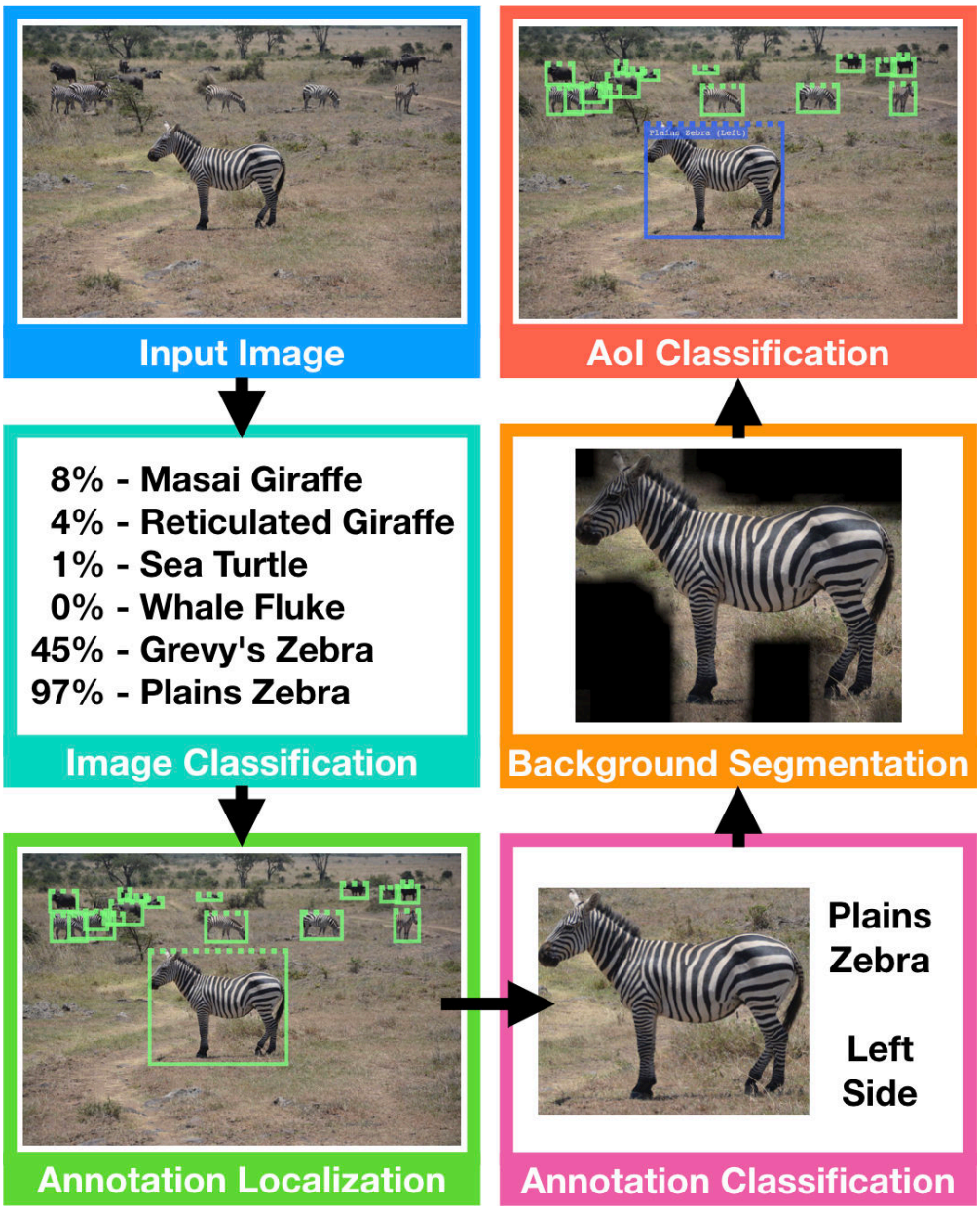


Figure 3.2: An overview of the detection pipeline and its components: 1) image classification provides a score for the species that exist in the image, 2) annotation localization places bounding boxes over the animals, 3) annotation classification adds species and viewpoint labels to each annotation, 4) annotation background segmentation computes a species-specific foreground-background mask, and 5) Annotation of Interest (AoI) classification predicts primary animal(s) of the image. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

their intended usage in the detection pipeline, the following: 1) whole-image classification to select the images that contain desired species or are otherwise relevant to the analysis, 2) bounding-box localization to form annotations, 3) annotation labeler to predict the animal's species and viewpoint, 4) coarse annotation segmentation to eliminate irrelevant background information, and 5) a classifier to select the "Annotation(s) of Interest" as the most prominent animal(s) in the image. We will also discuss additional components and use cases that add functionality and make the detection pipeline a more comprehensive solution, including detecting parts within an annotation or orienting annotations with a predicted rotation. We will start the discussion by reviewing two new datasets for animal detection that this research contributes.

3.1 Animal Detection Datasets: WILD & DETECT

This chapter presents two new detection datasets, called WILD (Wildlife Images and Localizations Dataset) and DETECT, alongside the proposed detection pipeline. The purpose of these datasets is to provide a more realistic collection of *in situ* animal sightings that often are not found in public detection datasets. The hope is that these datasets can motivate future ecological research on animal detection and provide examples of how best to curate datasets for the problem domain of animal re-ID.

3.1.1 WILD Dataset

The WILD dataset is comprised of photographs taken by biologists, wildlife rangers, citizen scientists [132], and conservationists, and captures detection scenarios that are uncommon in publicly-available computer vision datasets like PASCAL [127], ILSVRC [35], and COCO [128]. Furthermore, human photographers implicitly curated all animal sightings in the WILD dataset (i.e., a person actively decided to take the picture). This feature is in contrast to what would be seen in 1) movement-based camera trap, 2) exemplar-based datasets like iNaturalist[6], or 3) "always-on" overhead aerial surveys of animals. Unfortunately, most computer vision datasets make no distinction between wild animal sightings and other representations of that species (e.g., a stuffed zebra animal toy or animals seen only in captivity). Public challenge datasets are not representative of the types of images generally collected by park rangers and tourists, which is how image data for animal population censusing is gathered (discussed in Chapter 6). Class definitions that allow broad acceptance distract from the task of detecting real-world sightings of animals in the wild and are

[6] `inaturalist.org` (Accessed: Oct. 29, 2021).

generally incompatible with ID. In contrast, all of the images in the WILD dataset were taken of wild animals in their natural habitats.

The species cataloged by WILD are 1) Masai giraffe (*Giraffa camelopardalis tippelskirchi*), 2) reticulated giraffe (*Giraffa reticulata*), 3) sea turtle (*Chelonia mydas* and *Eretmochelys imbricata*), 4) humpback whale fluke (above-water flukes of *Megaptera novaeangliae*), 5) Grévy's zebra (*Equus grevyi*), and 6) plains zebra (*Equus quagga*). The WILD dataset offers challenging detection scenarios for each species; for example, zebras and giraffes tend to form social groups and stand close together, creating sightings with frequent bounding box overlap, occlusion, and cross-species co-location. It would be hard, or at the very least inefficient, to capture this level of complexity in artificial settings like a zoo – especially since a zoo setting largely omits the chance of seeing other species in the same image and will duplicate background textures. The common practice for sea turtles is to photograph the animal in and out of the water (if possible). As a result, there are two major modalities in the dataset for sea turtles: underwater backgrounds and more standardized backgrounds on land. The dataset also has examples of Humpback whale flukes to provide a contrasting species that is easy to detect but much harder to identify (contour-based ID). Finally, WILD has animal detections for two species of giraffe and two species of zebra, which must be distinguished. From an ecological and censusing perspective, it would be inappropriate to lump Grévy's zebra and plains zebra into a single "zebra" label because these two populations are distinct and may have radically different scopes of conservation concern. Experience with large computer vision datasets gives the general impression that classification labels are often much too broad to use as ground-truth for training an animal detection system for real-world applications on endangered species.

A dataset of 5,784 images was gathered, and 12,007 annotation bounding boxes were hand-annotated for 30 classes. The six species of interest that are the focus of the dataset have 9,871 annotations in total. A breakdown of the number of images and annotations that contain each species can be viewed in Table 3.1. The annotations were cropped out of the original images and were assigned to human reviewers to label the animal's species and viewpoint. Reviewers were then tasked to pick the most prominent annotation(s) in each image for Annotation of Interest (AoI) classification, which is discussed at length in Section 3.6. A total of 3,602 annotations were marked as AoIs (36.5%). The dataset was then partitioned into two sets: training (4,623 images) and testing (1,161 images) through an 80/20% stratified split that ensured the number of annotations per image was balanced across the split. This splitting results in a total of 7,841 annotations for training and

Table 3.1: The WILD dataset has 1,000 images for six different species. The total number of images is slightly less than 6,000 because some species share sightings within the same image, specifically between zebras and giraffes, demonstrating the need for a multi-prediction image classifier. There are also an additional 2,136 annotations in this dataset of miscellaneous categories (`car`, `boat`, `bird`, etc.). ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

Species	Images	Annotations	Annotations of Interest
Masai Giraffe	1,000	1,468	611
Reticulated Giraffe	1,000	1,301	595
Sea Turtle (Green and Hawksbill)	1,000	1,002	567
Whale Fluke	1,000	1,006	595
Grévy's Zebra	1,000	2,173	669
Plains Zebra	1,000	2,921	561
TOTAL	**5,784**	**9,871**	**3,598**

2,030 for testing. The dataset is distributed[7] in the PASCAL VOC format with additional metadata attributes to mark viewpoints and AoI flags.

3.1.2 DETECT Dataset

A more specialized dataset than WILD is also contributed, called DETECT, which is comprised of sightings of only Grévy's and plains zebras. The purpose of this dataset is to provide a more comprehensive real-world understanding of how these two species (with very similar visual appearance) are seen together and annotated for photographic censusing (discussed in Chapter 4). The DETECT dataset was constructed from 2,500 images taken by ecologists, field technicians, computer vision researchers, and volunteer citizen scientists [131], [133] working in Kenya. The images were taken such that the primary focus had to be one of these two species of zebra; bounding boxes and species labels were then annotated by hand, labeling plains zebra as `zebra_plains` and Grévy's zebra as `zebra_grevys`. In addition to zebras, bounding boxes were generated for other animals present in the images (if any) and assigned with an `unspecified` species label. The number of annotations per image was much higher for this dataset (zebras like to herd together in social groups compared to a solitary sea turtle).

[7] `https://cthulhu.dyn.wildme.io/public/datasets/wild.tar.gz` [1.4GB] (Accessed: Oct. 29, 2021).

Table 3.2: The number of viewpoints for each species in the DETECT dataset. An unbalanced distribution of viewpoints is due to 1) the behavioral characteristic of zebras and 2) the preference of field scientists in previous manual mark-recapture studies to photograph a single side. ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

Viewpoint	Plains	Grévy's	Unspecified	Total
`left`	**1,965**	565	120	2,650
`front-left`	226	116	29	371
`front`	83	69	32	184
`front-right`	104	137	17	258
`right`	424	**1,029**	147	1,600
`back-right`	168	326	36	530
`back`	190	244	36	470
`back-left`	381	186	25	592
Total	**3,541**	**2,672**	**442**	**6,655**

Finally, viewpoint information was annotated for each bounding box in DETECT by assigning it to one of eight views of the animal's body: `left` (L), `front-left` (FL), `front` (F), `front-right` (FR), `right` (R), `back-right` (BR), `back` (B), and `back-left` (BL). The entire dataset has 6,655 ground-truthed bounding boxes with 3,541 plains, 2,672 Grévy's and 442 "unspecified". The breakdown of viewpoints by species is shown in Table 3.2. A challenge to photographing real-world zebras is that capturing a balanced number of viewpoints can be difficult, with `front` being the least photographed in the dataset. The strong bias for the photos showing plain zebras with left-side viewpoints, and Grévy's zebras with right-side viewpoints, is due to historical reasons in the way animals were identified by-hand for manual mark-recapture studies [2], [358].

The dataset is available[8] as a Wildbook IA (WBIA)[9] database for training and evaluation. The data was split into subsets with 60% for training, 20% for validation, and 20% for testing. For all model training described in subsequent evaluations, the training and validation sets were combined, and the trained models were evaluated against only the test set. The partitioning of the various sets, while random, was balanced to respect both the distribution of species and the number of annotations

[8] `https://cthulhu.dyn.wildme.io/public/datasets/detect.tar.gz` [18.0GB] (Accessed: Oct. 29, 2021).

[9] `https://github.com/WildMeOrg/wildbook-ia` (Accessed: Oct. 29, 2021).

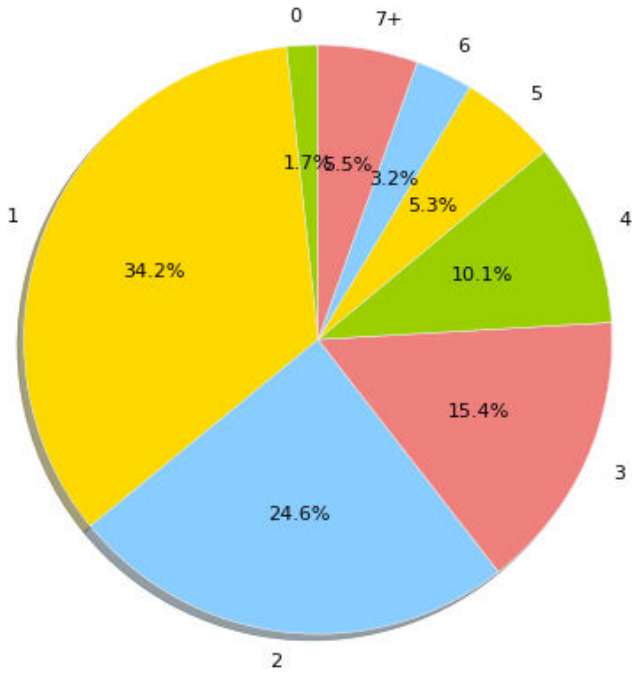


Figure 3.3: The distribution of densities (bounding boxes per image) in the DETECT dataset. A density of 0 indicates that the image was taken containing no animals. The maximum density of any image in the dataset is 23 but was capped at 7 (or more). ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

per image (see Figure 3.3 for a distribution). The viewpoint was not considered for this balancing procedure because many of the images contain multiple photographed animals photographed from differing viewpoints. A total of 501 images (with 1,343 ground-truth annotations) comprise the test set, which is for what the following results are reported.

The rest of this chapter will describe the core components of the detection pipeline in the order that they would be applied to an input image. The neural network models (except for the bounding box localizers) are trained using Lasagne [359] and Theano [360], [361] on a single NVIDIA TITAN V GPU with 12GB of video memory[10]. The localizer models are trained using 1) the original Caffe [363] source code for Faster R-CNN[11], 2) an original Python wrapper[12] for

[10]Much of this work was originally developed in 2016 and 2018. Since then, most of the detection pipeline's components have been modernized and re-implemented with PyTorch [362] and are on PyPI under `wildbook-ia`

[11]`https://github.com/rbgirshick/py-faster-rcnn` (Accessed: Oct. 29, 2021).

[12]`https://github.com/WildMeOrg/wbia-tpl-pydarknet` (Accessed: Oct. 29, 2021).

the original YOLO V1 source[13] by Redmon *et al.*, and 3) a PyTorch [362] re-implementation [14] by Ophoff *et al.* [364] of the original open-source implementation of YOLO v2. The Python bindings[15] for the Hough Forests implementation by Gall *et al.* [114] are used for this evaluation.

3.2 Whole-Image Classification (WIC)

The Whole-Image Classifier (WIC) is the first stage of the proposed detection pipeline. Its primary goal is to make a "relevancy check" for all of the processing in the detection pipeline. Thus, for example, there would be no need to power up an advanced animal detection neural network and load its hefty weights into GPU memory for an image of a birthday party.[16] A benefit of using a first-pass image classifier is that – due to advances in neural networks (see Chapter 2) – it can be trained with relatively little training data. We can therefore expect this component to be very accurate and fast. We analyze two use cases where the WIC is helpful for an automated animal detection pipeline: 1) checking for the existence of relevant species in an image for further processing, and 2) quickly processing large amounts of images to eliminate trivial negatives (e.g., filtering out false triggers made by camera traps).

3.2.1 Species Existence Classifier

One of the most common purposes of the whole-image classifier (WIC) is to quickly predict the existence of species of interest within an image. Unlike the original ILSVRC classification challenge that offered only a dominant whole-image class with 1-class and 5-class testing modes, there is often a need to classify images containing multiple animal sightings for more than one species (of equal importance). This distinction is important because some animal species (e.g., Grévy's zebra and reticulated giraffes) have overlapping migratory ranges and are sometimes seen together in images. Therefore, the WIC is designed to predict a multi-prediction, multi-target vector. The vector's corresponding index is set to 1.0 if at least one animal of that species exists in the image and 0.0 otherwise. Note that this network is not tasked to count the number of animals for a given species in the image. Instead, it simply needs to produce a *true* or *false* flag for if the species exists.

The network takes as input a 192×192-pixel image that is reduced to a 5×5×128 feature

[13] `https://github.com/pjreddie/darknet` (Accessed: Oct. 29, 2021).
[14] `https://github.com/WildMeOrg/wbia-deprecate-tpl-lightnet` (Accessed: Oct. 29, 2021).
[15] `https://github.com/WildMeOrg/wbia-tpl-pyrf` (Accessed: Oct. 29, 2021).
[16] This is an actual situation that was encountered when the contents of an SD card was copied during a census.

vector via convolutional and max-pooling layers. The network then adds a 256-dimension dense layer, followed by a feature pooling layer, a dropout [41] layer ($p = 0.5$), and another 256-dimension dense layer. The final dense layer has six outputs, one for each species of interest, and uses a sigmoid activation function. The design of the image classifier purposefully does not normalize the network's output with a softmax activation function, nor does it penalize for categorical cross-entropy loss. As a result, the output is not a 1-hot vector and does not produce a discretized PDF that sums to 1.0. For example, a valid network output could predict the existence of all (or none) of the target species in a given image.

For all of the neural network classifiers in the detection pipeline, extensive data augmentation is employed during training to help control over-fitting. Data augmentation aims to provide the network with a slightly different training example for each epoch, sampled such that the mini-batches are also ordered randomly. The augmentation is applied at runtime each time the training example is loaded into memory and not simply applied beforehand and cached to disk, saving disk space and increasing the amount of randomization during training. The augmentation performs the following operations, each randomized for every training example:

1. exposure in the `Lab` color space on the luminance channel
2. slight hue shifts,
3. rotation, scaling, and Affine skewing,
4. horizontal flipping (generally no vertical flipping), and
5. blurring.

The WIC model does an excellent job at correctly predicting species existence within an image, as shown in Figure 3.4. The worst-performing species (Masai giraffe) achieves a ROC area-under-the-curve (AUC) of 96.3%, and the best-performing species (whale fluke) has an almost-perfect 99.94% AUC, missing only a handful. The mean AUC across all species is an outstanding 98.3%. The operating points were selected as the closest values on the curve to the top-left corner (indicated by the colored dots on each curve), providing the optimal AUC and balancing the true-positive rate (TPR) against the false-positive rate (FPR). With optimal points selected independently for each species, the image classifier can be used to predict an existence value for all six species simultaneously. When we consider this use case, the classifier is correct 64.8% of the time at predicting the exact X-hot vector for a given test image. Nevertheless, the vast majority of the errors have a Hamming distance of 1, meaning the algorithm incorrectly predicts only one of the values in the vector. Over 90% of errors are between the two species of giraffe and zebra, predicting

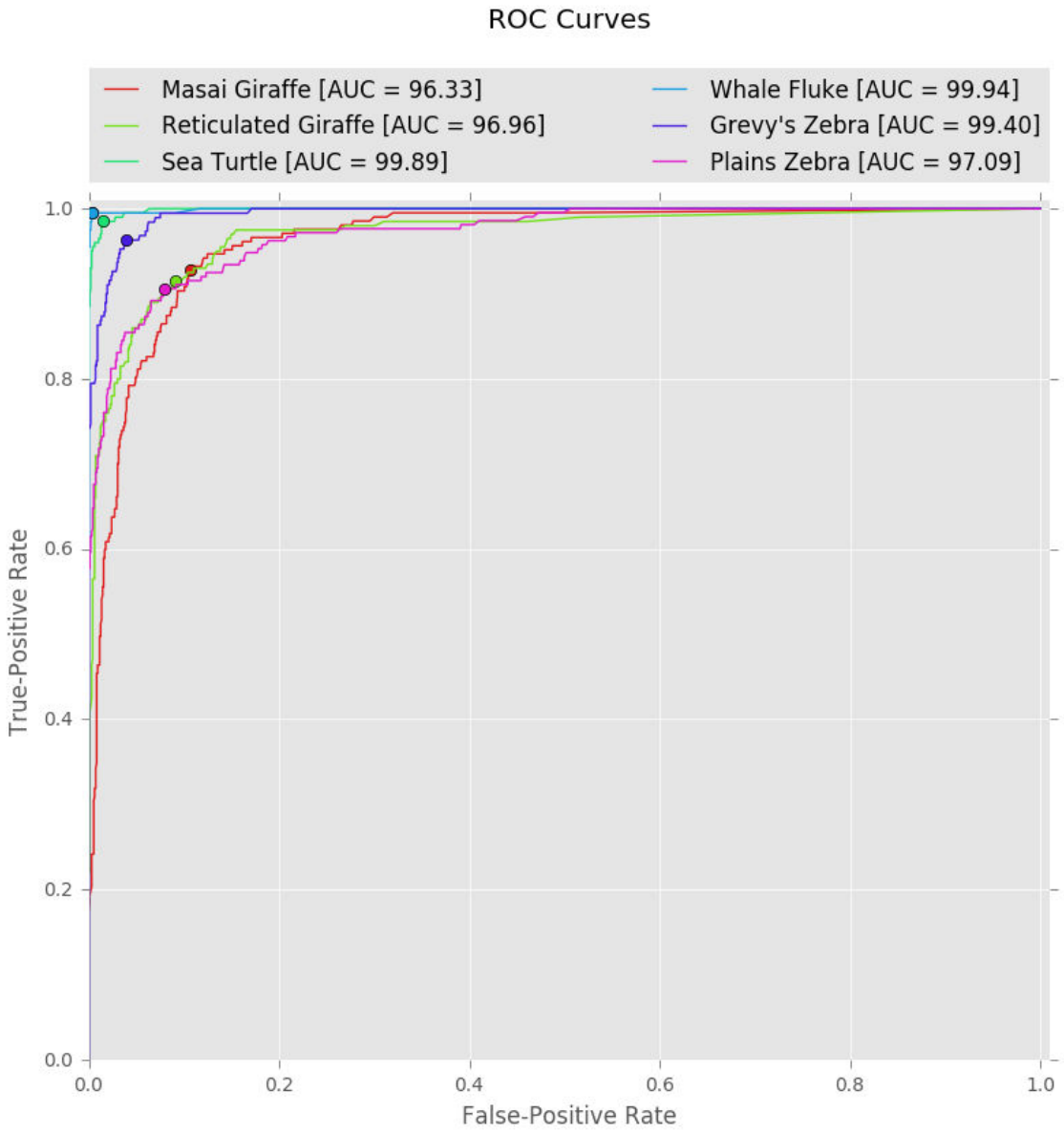


Figure 3.4: The ROC performance curves for the Whole Image Classifier (WIC). We can see that the WIC component of the detection pipeline performs extremely well on all species. Some species are harder than others, notably giraffes and plains zebras; this error can be attributed to the similar appearance of the giraffe species, leading to confusion. All species have an impressive AUC greater than 96%, making it an accurate first-pass filter for the detection pipeline. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

existence for both when only one is shown in the image. Since the rest of the detection pipeline can distinguish between similar species, these errors are not a major concern because they should be caught by later, more focused (working with annotations, not images) components of the pipeline.

Using the WIC as a content filter is technically optional; we could choose to rely on the outputs of the second component, the localizer, to perform the initial content filtering. However, preliminary results by Beery *et al.* [115] suggest that a localizer will slightly outperform a whole-image classifier at the filtering task on camera trap data but at the cost of much higher inference times. In addition, the WIC has significantly fewer parameters than the localization networks presented in the next section, which makes it much faster to run on large volumes of images. For comparison, the WIC (run on batches of 1024 images) can return results in approximately 3-4 seconds using GPU acceleration. In contrast, localization networks need to run smaller batches of around 128 (for memory reasons) and take upwards of 30 seconds (real-time) on the same GPU and images. Therefore, the benefits of using a WIC with localization compared to using just a localizer can be profound when computational resources or runtime requirements for the application are limited.

3.2.2 Filtering Camera Trap False-Alarm Triggers

Another application for the whole-image classifier is to use it as a fast binary classifier to filter out irrelevant images. One example of where this function is valuable is very quickly processing raw images taken by a camera trap or aerial survey. Images collected by a motion-triggered camera trap are expected to have a high ratio of false positives because the trigger is not context-aware (i.e., images taken that do not contain any sightings of desired species). To set the scale of this problem: the WIC classifier could, for example, be used to search through 500,000 camera trap images to find less than 1,000 that had sightings of aquatic jaguar (*Panthera onca*). The sheer size of the problem is an issue for running slower localization algorithms on this amount of data and generating sufficient training data to train the WIC (or localizer) on novel camera trap collections with minuscule true-positive rates.

Fundamentally, we can re-formulate the WIC as a simple binary classifier (2 classes with standard cross-entropy loss) that predicts either “keep” or “discard” for a given image. The benefit of a binary design is that it can be very flexible, allowing a camera trap researcher to decide the usefulness of an image in whatever way is needed. A web interface was designed to give a randomized example to a reviewer for annotation, as shown in Figure 3.5. The benefit of this tool is

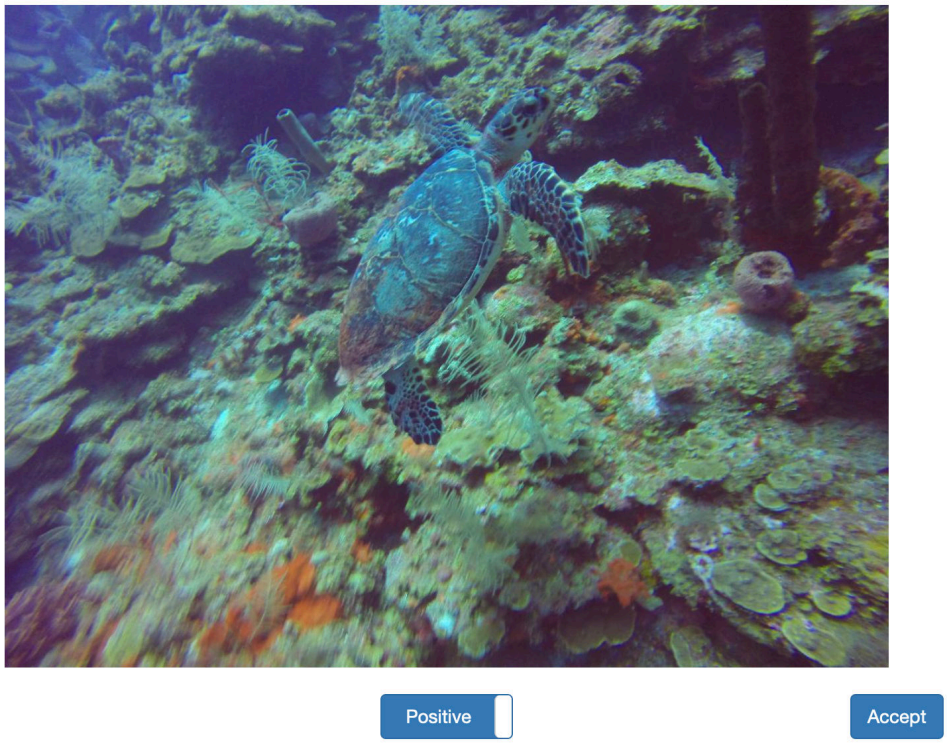


Figure 3.5: The web interface for reviewing Whole Image Classifier ground-truth labels. When using the WIC in binary mode, a simple flag on the entire image can be assigned for "keep" (positive) or "discard" (negative). When run in the multi-prediction, multi-target mode, the localization web interface and subsequent annotations are used as the WIC's ground-truth training labels.

Figure 3.6: Example camera trap images of true-positive (left, animals detected) and false-positive (right, nothing of interest) triggers, taken from two camera-trap datasets.

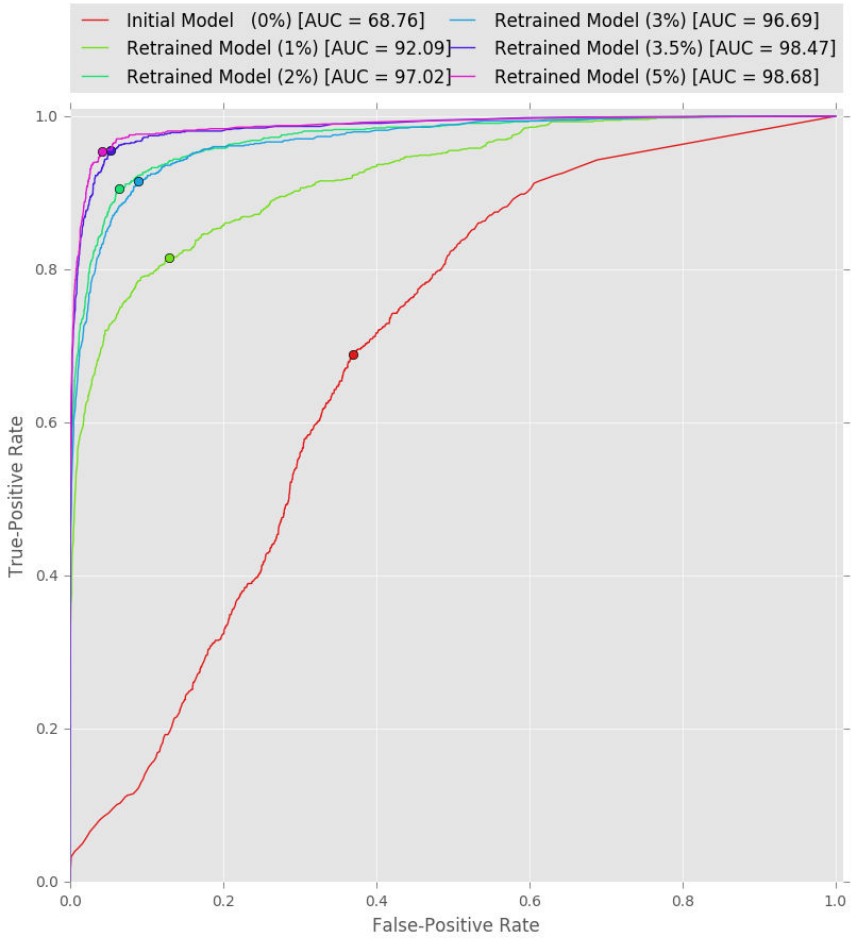


Figure 3.7: The ROC performance curves for the Whole Image Classifier on camera trap photographs. The best model with 5% of the data annotated achieves a classification accuracy of 96.5%.

that it allows a researcher to quickly create ground-truth for training the WIC component that is tuned for a unique camera trap location and survey time, often to create a one-time-use ML model. One of the apparent benefits of using the WIC as a filter for camera trap images is that it does not require extensive (and relatively labor-intensive) bounding-box training data. If available, the network can use bounding boxes to specify existence ground-truth (essentially a "yes" or "no" value for keeping an image), but this is not a requirement. As such, gathering a large amount of training data for the WIC is manageable and efficient because it can be trivially distributed to multiple human annotators.

The results of the WIC are presented here on a proprietary dataset captured by Dr. Megan McSherry at Princeton University using motion-based camera traps in Kenya. While state-of-the-art

results on this task are not reported here, a benchmark of the WIC classifier is presented for future baseline comparisons. We refer the reader to the work of Beery *et al.* [115] for a more advanced framework on how to detect animal movement in camera traps. The dataset was captured with camera traps placed in specific areas to photograph domesticated grazing of herding animals by local tribes. Figure 3.6 gives an example of a positive and negative image – we see that (left) shows sightings of sheep and (right) is an empty field with nothing of interest (a false trigger). An initial model was trained using a baseline ground-truth split between a few hundred positive and negative images provided by the researchers (referred to as the 0% split). New WIC models were then trained using hand-annotated ground-truth data, in approximate intervals of 1% of the total size of the dataset, up to 5%. Figure 3.7 shows the performance of each WIC model as more ground-truth examples were provided. Even with only 1% of the entire dataset annotated, the WIC model can achieve an impressive Area Under the Curve (AUC) of 92.1%. Annotating a total of 5% of the data results in a final classifier that achieves 98.7% AUC on held-out validation data. However, we can see that the performance increase with more ground-truthed training data quickly diminishes; effectively, the dataset could have only been annotated to 3.5% (98.5%) for approximately the same performance level as 5% (98.7%). These results show that the WIC can be a highly effective and accurate tool for filtering camera trap imagery while putting light expectations on data annotation. The evaluation focuses on small amounts of training data. It shows that annotating even a 1% random data sampling can perform as a weak classifier to eliminate many false-alarm images (92.1% AUC).

After the WIC is applied to the raw images and filtered appropriately, the next step of the detection pipeline is to localize all of the relevant animals in the images. This step is crucial because images may contain multiple animals, or the animal could be small relative to the size of the image. Therefore, some process is needed to convert images into annotations of distinct animals for the identification procedure.

3.3 Annotation Bounding Box Localization

The second component of the detection pipeline is tasked with generating bounding boxes and species labels for the relevant animals in an image. Localization is vital from an identification perspective because it allows for the separation of distinct animals, gives a consistent and comparable scale of the different sightings, and supplies a method (cropping) to remove large amounts of distracting background information. Furthermore, the preciseness of the bounding boxes (i.e.,

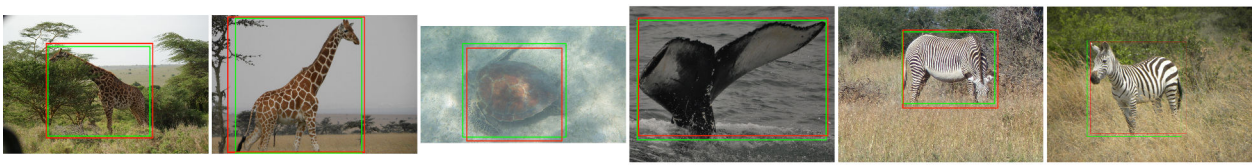


Figure 3.8: Example annotation localization predictions on single-sighting exemplar images for each of the six species of interest in the WILD dataset. The green boxes designate ground-truth bounding box coordinates, and the red boxes represent the localization bounding box predictions. Since annotation classification is also performed, these bounding boxes are treated more like salient object detections. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

how well they fit snugly around an animal) can play a critical role in the overall accuracy of the identification pipeline, something that will be examined in-depth in later chapters. After all, annotations are the primary lens through which ID sees the world, and the localizer should be very careful not to produce confusing or poorly-formed bounding boxes. This filtering role also means that the quality of the localization component plays a crucial role in the overall quality of a photographic census. Hence, it is a significant focus of evaluation in this chapter.

Two fundamentally different detection techniques for the localization component are compared: a pre-neural network approach and (two) neural network approaches. As perplexing as it may seem in the era of deep learning to use anything but a neural network, random forest detectors are easy to train because they have fewer hyper-parameters, built-in regularization, and require much less data to train. Furthermore, they do not rely on heavy GPU-accelerated computation for training, making them an ideal candidate for bootstrapping when few training images are available. The Hough Forests [106] variant of random forests, in particular, is resilient to partial and occluded objects due to its voting scheme [107]. As such, a random forest-based detector and two CNN-based detectors (Faster R-CNN [29] and YOLO v2 [196]) are evaluated on the WILD Dataset.

Before we begin, we need to review exactly how a bounding box is defined. A bounding box i for a given image I is parameterized by the following equation:

$$\text{bbox}_i(Image, x, y, w, h, \theta) = \text{rotate}(I[y : y + h][x : x + w][:], \theta) \tag{3.1}$$

where cropping is applied before rotating around the center of the bounding box. While the localization algorithms presented in this section do not apply any rotation directly (and only predict axis-aligned bounding boxes with species labels), a separate orientation component can rotate and fix the boxes if needed. The orientation task is a separate module because it allows the localizer to be replaced without requiring the new algorithm to support rotation (a relatively rare need). It is also worth noting quickly that not all animals in an image are worth localizing. For example, small birds or very distant animal herds are ignored in this evaluation because they are categorically not the focus of censusing events, especially since it can be exceedingly tedious to annotate ground-truth thoroughly. Thus, while ground-truth completeness in bounding boxes is crucial, there is a point where it is simply not realistic to perfectly annotate (or automatically find) every single animal in an image. Examples of easy object localizations for each of the six species in the WILD dataset can be viewed in Figure 3.8.

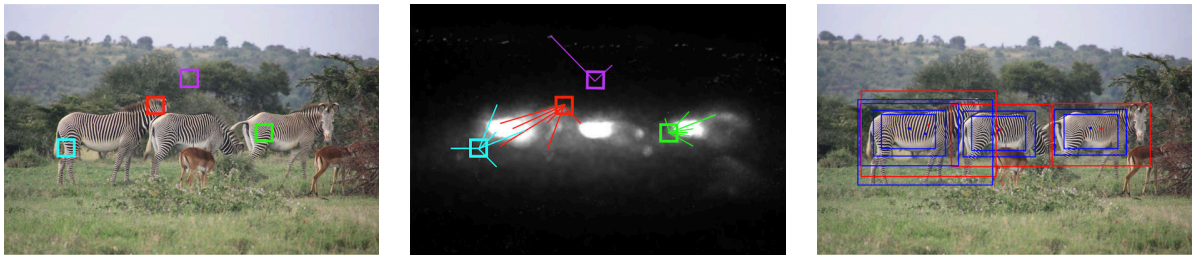
3.3.1 Hough Random Forests (RF)

Hough Forests are an ensemble of random binary trees. Each tree attempts to optimize the performance of classification and regression by performing a series of binary pixel tests; the authors of [106], [365] demonstrate that training a random forest tree in this combined fashion benefits both generalization and accuracy. The first pre-processing step of training extracts a collection of small image patches (32×32 pixels) from the ground-truth to compose a large set (60,000) of positive and negative training patches. Importantly, each positive patch records its relative offset to the center of its corresponding object in the ground-truth.

During training, each tree is given the same set of patches. The implementation used in this evaluation has an ensemble – hence, a "forest" – of 10 trees. Each tree generates tests that split the patch dataset at each non-leaf node, which performs a random binary pixel test on each image patch P. The test, as formulated in [114], is

$$test_{\alpha,p,q,r,s,\tau}(P) = \begin{cases} 0, & \text{if } P^{\alpha}(p,q) < P^{\alpha}(r,s) + \tau \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

where α is the channel of the image patch, (p, q) is a random location in the patch, (r, s) is a different random location in the patch, and τ is a threshold offset. Every node is allowed to pick a



(a) Source Image (b) Hough Image (c) Final Detections

Figure 3.9: Hough Forests test patches are extracted densely over a test image (a) and are classified using an ensemble of random binary trees into a collection of leaves. Each leaf has a set of positive and negative patches given to it during training, which are used to make weighted probabilistic Hough votes into an aggregate Hough image (b). The high-probability object center peaks (white) are used to generate bounding box proposals (c, blue). The proposals are filtered with non-maximum suppression to create the final detections (c, red). Compare the votes of the red and purple test patches in (a, b); the purple votes are sporadic and do not accumulate, whereas the red votes contribute to an object center. The blurring on peaks is due to voting confusion. ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

unique randomized test, randomly seeded to promote diversity. Every node samples randomly over the parameters α, p, q, r, s, τ to find the best binary test that minimizes either the positive-negative classification error or the positive-patch regression error. The node will minimize (with $p = 0.5$) either the binary cross-entropy classification error or the regression offset sum-squared difference error (negative patches are ignored since, by definition, they have no center offset to an object center). Each tree is constructed recursively to a depth of 16 layers during training or creates a leaf when a node has fewer than 20 patches.

During test time, each leaf node holds a collection of positive and negative patches. A leaf's positive class probability is computed as the percentage of positive patches out of all patches it received by the end of training. Each positive patch in a leaf makes a weighted probabilistic vote for the object's center in the test image based on where that patch originated in its respective ground-truth image. As shown in Figure 3.9, these Hough-transform votes (a) are computed densely across the entire test image and aggregated over multiple scales to generate a combined Hough image (b). The bright white spots in the Hough image indicate the probable locations of object centers. Thresholded peaks are selected as candidate center proposals, object bounding boxes are

derived from the locations of patches that voted for the particular peaks (c, blue), and non-maximum suppression is applied to produce the final detection regions (c, red).

The Hough Forests detector has some distinct advantages: 1) it is easy to parallelize across multiple CPU cores for efficient training and inference processing, and 2) the voting scheme will aggregate probabilities originating from *any* location on an animal. For example, if only the face and neck of a zebra are visible in an image, the face and neck tree leaves will still make probabilistic votes for where it thinks the center of a zebra should be, even if that location is off the edge of the image or is occluded. This voting scheme makes Hough Forests more resilient to occlusions and makes it an attractive solution to the challenges present in the WILD dataset. However, the image patches are too small to learn precise localization information (i.e., a zebra neck patch can look very similar to a zebra hip), which results in a distinct blooming effect of voting confusion surrounding an object's center in the Hough image. Moreover, the implementation of Hough Forests used here is trained as a binary classifier (one-vs-all) and, therefore, cannot natively represent multiple classes within the same tree. This limitation poses a problem with representing multiple poses of the same species, as some views have conflicting spatial representations for an object's center. This conflict confuses training and detection, which hurts accuracy.

The evaluated version of Hough Forests improves on the efficiency and accuracy reported in [114]. The implementation adds OpenMP [366] multi-CPU parallelization, adds new image channels, supports multiple resolutions, drastically increases the number of binary tests performed at each node during training, and makes more intelligent bounding box regression decisions with the coordinates of voting patches. The details explained in this section are meant to augment the algorithmic summary in previous work [2], which used Hough Forests to detect plains zebras and Masai giraffes in photographs taken at the Nairobi National Park in Nairobi, Kenya.

3.3.2 Faster R-CNN

The Faster R-CNN network by Ren *et al.* [29] is the third iteration of the R-CNN approach introduced by Girshick *et al.* [24], [190]. Each new iteration of this detector family has a more simplified training process, speed improvements, and improved accuracy. For these reasons, there is little benefit in evaluating the preceding R-CNN [190] and Fast R-CNN [24] algorithms in this discussion. The motivation behind Faster R-CNN is that the Selective Search [367] candidate proposal phase used by its precursors is a significant performance bottleneck. The authors re-implement the bounding box candidate proposal as a neural network and brand it as a Region

Proposal Network (RPN) to address this problem. The critical insight to training Faster R-CNN is that the RPN is a separate network from the classification network inherited from [24], but – to reduce processing – the two networks share most of their convolutional filters. The RPN and classifier are given the same fixed proposals during training, and the two networks alternate back and forth to update the weights.

During test time, the shared convolutional features are only computed once. The RPN adds additional convolutional layers on top of the shared filters to generate localization predictions. The benefit of this architecture is that the training is mostly unified, which dramatically increases speed performance during both training and testing. Furthermore, replacing Selective Search with the RPN also improves accuracy. However, the branching top of the network involves additional complexity during training, and the network still does not quite achieve real-time performance on GPUs. The Faster R-CNN network runs at about six frames-per-second on high-end GPUs but achieves state-of-the-art performance for detection [35] as of 2016. The implementation of Faster R-CNN seen in this evaluation is unmodified other than training for 30,000 iterations (with the newer and faster "end-to-end" scheme) on different classes. Unfortunately, the training diverged several times before a stable model was produced because the RPN failed to generate valid bounding boxes.

3.3.3 You Only Look Once (YOLO)

The You Only Look Once (YOLO, version 1) network by Redmon *et al.* [28] is a variant of single-shot detectors (e.g., SSD [167]). Single-shot detectors directly output a fixed-length regression output for a given fixed-sized input image without needing a separate region proposal network (RPN). Refer to Section 2.2.4 for more details. The architecture of YOLO is somewhat unusual as it uses a relatively large input image (448×448 pixels, compared to its contemporaries that mostly use 224×224 pixels) and produces 98 detection regions from a grid of 7×7 classification cells. Therefore, the network's output is always 98 bounding box coordinates along with an object score for each of the six species classes from WILD on each bounding box.

The YOLO detector implements a truly unified network architecture. The network produces multi-class bounding box candidates directly from a single forward inference on an image. See Figure 3.10 for a high-level comparison between Faster R-CNN and YOLO. The benefit of a unified integration is most notably speed for the cost of a slight drop in accuracy. For YOLO, the re-sized training images are given to the network in batches of 64, and the error gradient for each of the

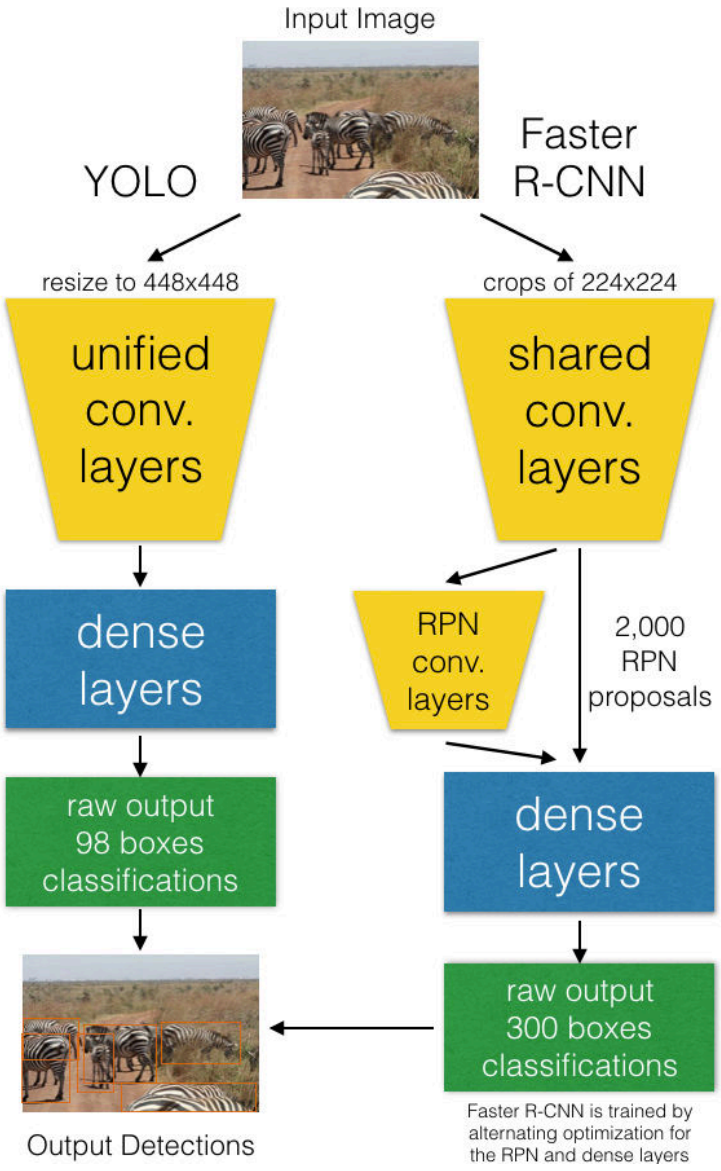


Figure 3.10: The YOLO network is a unified architecture that is trained top-to-bottom to minimize bounding box regression and classification error. In contrast, Faster R-CNN has a separate Region Proposal Network (RPN) that proposes salient object bounding box proposals, which are classified to produce class probabilities. Faster R-CNN is trained by alternating the training between the RPN and the classification "networks" until it converges, applying both gradients to the shared convolutional layers. ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

98 network detections is computed directly using the ground-truth bounding boxes, which are mapped onto the range $[0, 1]$. The entire image is given to the network during test time, which outputs a vector encoding of the 98 bounding box coordinates, a saliency probability for each bounding box, and class probabilities for every 49 (7×7) classification cell. Thus, each 64×64-pixel classification cell contributes two bounding box proposals. The bounding box saliency probabilities are combined with the corresponding cell's classification probabilities to create the final class probabilities assigned to each bounding box. The class with the highest probability becomes the bounding box label, and final detections are generated by thresholding the low scoring probabilities and non-maximum suppression.

The YOLO network is trained to optimize a complex, multi-part loss function. The mathematical definition of the loss function is presented in [28], but – to summarize it here quickly – the loss has five components: 1) a regression sum-squared difference loss (SSDL) for each cell's bounding box center x and y pixel, 2) a regression SSDL for the square-root of each bounding box width and height, 3) a conditional SSDL for the saliency probability of whether an object exists in a bounding box, 4) a corresponding conditional SSDL for if an object does *not* exist, and 5) an SSDL for the class probabilities of each cell. [28] To combat training instability, the authors use 1) two weighting hyper-parameters on the regression and classification loss terms to balance their respective error gradients, and 2) a unique learning rate schedule (called "burn-in") that starts intentionally very slow, then increases around iteration 600 for routine training.

YOLO V2 (version 2) [196] includes specific improvements to increase the network's ability to detect small objects more accurately. The network includes training improvements like batch normalization [73], multi-scale training and model improvements like anchor boxes, direct bounding box regression (instead of predicting residuals), is fully convolutional, and has a higher-resolution convolutional output. While these improvements helped stabilize training, the network still diverged several times before a stable random initialization was chosen and the model converged (trained for 24,000 iterations). Since YOLO V2 consistently outperforms YOLO v1, we do not report it in this evaluation and focus on comparing the detection performance of Hough Forests, Faster R-CNN, and YOLO v2.

3.3.3.1 Performance Trade-Offs

The YOLO network has distinct advantages over Hough Forests: 1) it has significantly more parameters to fit the training data, 2) has a larger effective receptive field for better regression

performance, 3) uses convolutional feature extraction with transfer learning on ILSVRC, and 4) is inherently multi-class. In comparison to Faster R-CNN, YOLO achieves real-time performance and, as mentioned previously, simplifies the entire detection pipeline down to a single forward inference. However, YOLO is more difficult to train with its poorly-behaving error gradient and has several network-specific hyper-parameters. Like Faster R-CNN, the YOLO network (realistically) requires a GPU to train, and both take significantly longer to train over Hough Forests. Both neural networks were trained for about 24 hours using two Titan X GPUs, whereas the ensemble of 10 Hough Forests trees was trained in just under 3 hours on a quad-core CPU. That being said, the testing speeds of both neural networks are at least two orders of magnitude faster compared to the Hough Forests implementation, which runs at roughly 15 seconds per image for nine scales.

On top of training speed advantages, the Hough Forests implementation does not require nearly as much training data. Furthermore, the convolutional filters of each neural network are initialized with pre-trained weights [46] before fine-tuning on the dataset. Between the two deep learning detectors, the YOLO network utilizes empty images (images with no ground-truth bounding boxes of any species) during training with implicit negative mining, whereas Faster R-CNN was not. This training procedure allows YOLO to see slightly more images during training, which represents 1.7% of the dataset (see Figure 3.3, density 0).

3.3.4 Results

The three detectors were evaluated by calculating the IOU (intersection over union) percentage between the detections and the ground-truth. A detection was considered correct if 1) the bounding box IOU ≥ 0.5 and 2) the species classification was correct. A classification error is when the IOU threshold was satisfied for a predicted bounding box but had an incorrect species label. Otherwise, the annotation was marked as having a localization error (not detected at all). Looking at Table 3.3 showing detection performance on the DETECT dataset, we can see that Hough Forests makes by far the most classification and location errors. The combined YOLO network achieves the highest number of correct detections overall but makes the most classification errors compared to the other two algorithms; Faster R-CNN makes more localization errors but rarely makes an incorrect classification. Hough Forests makes the fewest classification errors, but this can be deceiving since it has almost double the number of localization errors as YOLO. Side-by-side example detections for all of the algorithms can be seen in Figure 3.11. Overall, the YOLO V2 detector has the highest number of correct detections (56.0%) compared to Faster R-CNN (51.7%) and Hough Forests

Table 3.3: The number of correct detections and incorrect detections for two failure modes (localization and classification) of each algorithm, combined for both species. Localization errors fail to put a bounding box around an animal, while classification errors have a correct box but the wrong species label. The YOLO network gets the highest number of correct detections but has significantly more classification errors than Faster R-CNN. Faster R-CNN, while it makes more localization errors, seldom guesses the incorrect species. There are 1,343 test ground-truth detections (714 of plains, 536 of Grévy's, and 93 unspecified). ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

Algorithm	Localization Errors	Classification Errors	Correct
Hough Forests	59.6%	**0.2%**	40.2%
Faster R-CNN	47.8%	0.5%	51.7%
YOLO v2	**31.0%**	13.0%	**56.0%**

(40.2%) on the DETECT dataset, on top of being the fastest. Therefore, YOLO V2 is selected for all further analyses of detection performance.

The YOLO localization model has different performance curves for each species in WILD. The YOLO detector achieves a detection Average Precision (AP) of 57.6% for plains and 76.2% for Grévy's, as calculated by the area under a Precision-Recall curve. The whale fluke and sea turtle localizations achieve an AP of 99.0% and 93.5%, respectively. This high level of performance makes intuitive sense because a mostly rigid animal sighted against a stark background of the sea, ocean floor, or sky will be easier to localize than a compact herd of overlapping, occluded animals. As displayed in Figure 3.12 (left), the difference in difficulty can be seen noticeably in the relatively poor performance of the plains zebra localizations at only 57.5%. By referencing Table 3.1 we can see that the ratio of easy-to-find annotations (Annotations of Interest) to all annotations is the lowest at 19.2% for plains zebras compared to the average of 42.2% for all species. Furthermore, the ratio of annotations per image for plains zebra is the highest at 2.9 compared to the average of 1.6. Nevertheless, the YOLO localizer achieves an mAP of 81.7% across all species, suitable for generalized detection.

The performance of the localizer is further analyzed when only annotations marked as AoIs are considered, see Figure 3.12 (right). AoIs should be distinguishable, relatively large, and free of significant occlusions (a formal definition of AoI is provided later in Section 3.6). The annotation

Figure 3.11: Example images of detections on a set of 20 images for plains zebra (PZ) and Grévy's zebra (GZ). The operating point was set to 0.8 for the CNNs and 0.6 for Hough Forests (HF). ©2016 IEEE. Reprinted, with permission, from: J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

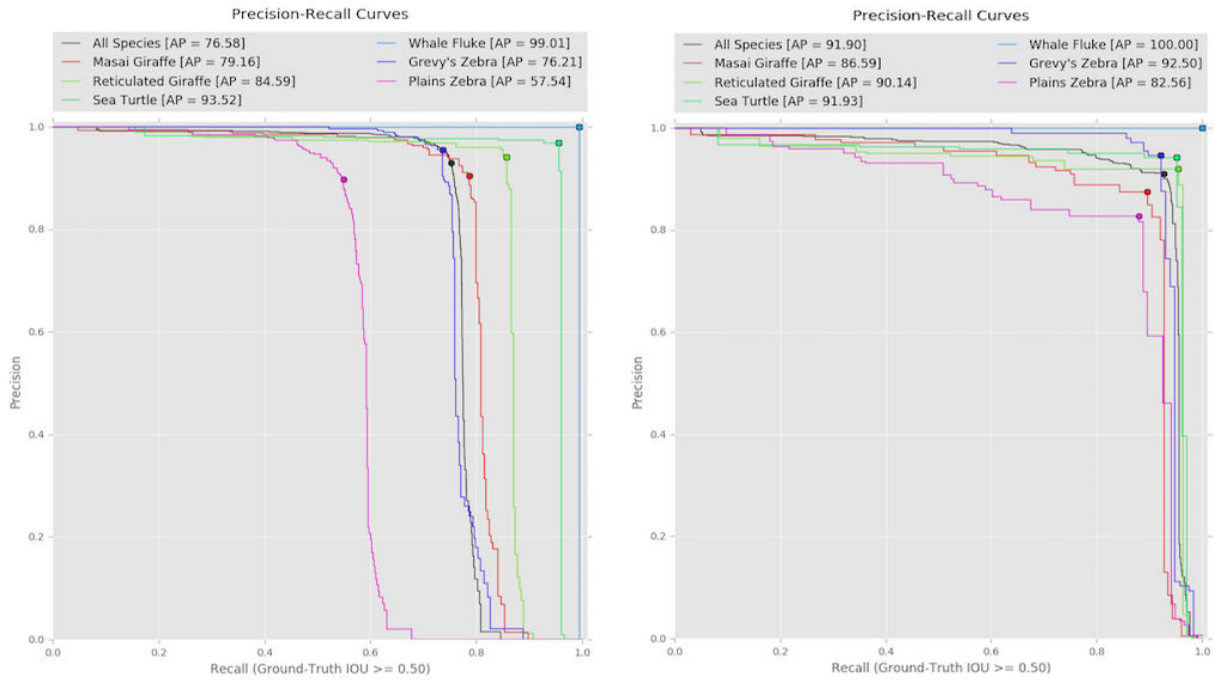


Figure 3.12: The annotation localizer precision-recall curves (left) reports an unfiltered mean average-precision (mAP) of 81.7% across all six species with an Intersection-over-Union (IoU) threshold of 50%. The drastic drop in performance of the plains zebra species can be contributed to the high number of background – likely small-sized – annotations for this species; focusing on just AoIs (right) increases mAP to 90.6%. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

localization performance drastically improves the detection Recall for all species when only AoIs are considered, with the most improvement being achieved by the plains zebra localizations. This improvement indicates that most of the localization errors – across the board for all species – are from the background, small, occluded, or otherwise unidentifiable animals. Missing these detections is less of a concern because we do not care to process them with ID. By adjusting the goal of the localizer to focus on maximizing performance for AoIs, the YOLO model can achieve an mAP of 90.6% on the WILD dataset, a significant improvement.

In summary, the creation of annotations by the localizer sets the context for which areas of an image are likely identifiable. Furthermore, the goal of an identification process is to match *comparable* sightings of animals; one of the most primitive pieces of information for determining if two annotations are comparable is their species. For example, it would not make sense to try and compare a zebra to a giraffe because the match will always have a predictable (negative) result.

However, as discussed next, knowing the species is not the only kind of ecological metadata helpful in determining if two annotations are comparable.

3.4 Annotation Labeling

The third component of the detection pipeline is designed to re-classify the annotations produced by the previous localization stage. The primary purpose of the annotation classification network (also known as the "annotation labeler") is flexibility. For example, the species-only classification output of the localizer may not be the final (or only) intended metadata for an annotation. Furthermore, it may be impractical to retrain the entire localizer when a new classification need comes up. A practical use case of the labeler is that it allows the pipeline to predict a species *and* a viewpoint for annotations. This function is handy for identification because knowing the viewpoint of an animal allows for incompatible annotations to be filtered out, even more so than compared to only filtering on species. It also allows for the localization network to be trained at a different level of abstraction when considering the ground-truth labels. For example, we may train the localizer to focus on a general "zebra" class (e.g., to optimize localization performance) but use the labeler to re-classify annotations as "Grévy's zebra" or "plains zebra", and add viewpoint classification support.

The labeler network uses smaller 128×128-pixel images as input; the annotation bounding boxes from the localization network are cropped out of the original image and are re-sized to the correct input dimensions. Input images are reduced to a 5×5×256 convolutional feature vector for classification via convolutional, max pooling, and batch normalization [73] (BN) layers. The network then adds a 512-dimension dense layer, followed by a feature pooling layer, a Dropout layer ($p = 0.5$), and another 512-dimension dense classification layer. The species and viewpoints are combined into paired classifications for the last dense layer of the network (activated by softmax), and the model's weights are optimized using the standard categorical cross-entropy loss.

The annotation labeler's architecture is similar to the WIC component, except it performs a standard single-prediction, multi-target classification. In addition, a separate set of weights (also initialized with transfer learning) is intentionally trained for the convolutional feature extractors in the WIC and labeler detection pipeline components. This separation increases redundancy but also allows for specialized filters to be learned for each task; each detection component can be independently optimized without needing to re-validate the performance impact of a unified feature extraction across the entire pipeline. Another reason the convolutional filters are not shared across

components is that the input image sizes are fundamentally very different since they are trying to capture different levels of detail. For example, the WIC is tasked with animal existence at an image level. In contrast, the labeler is sometimes tasked with differentiating similar species and viewpoints at an annotation level.

3.4.1 Results

The annotations are labeled with a viewpoint of the animal relative to the camera. The viewpoints for zebras and giraffes in the WILD dataset are labeled with one of 8 discretized yaw locations around the animal, from the set {`front`, `front-right`, `right`, `back-right`, `back`, `back-left`, `left`, `front-left`}. Sea turtles are commonly captured from above and sometimes from below, so their allowed viewpoints are constrained to the set of six viewpoints {`front`, `right`, `back`, `left`, `top`, `bottom`}. Whale flukes also have a similar restriction where they are label from a set of 4 {`top`, `bottom`, `right`, `left`}, with the most common being `bottom` when the angled fluke is viewed above water. The species and viewpoints pairs are combined into 42 distinct combinations (in the form `species:viewpoint`) to create the set of available classification labels for training. The label pairing used by the annotation classifier does cause an inherent class imbalance, but achieving balanced viewpoints across all species in a real-world, unstructured setting is an impractical goal. The real-world implication is that balanced training data for all categories is seldom possible when viewpoints are considered. Hence, the labeler needs to have some mechanism to counteract its effect on training. The labeler addresses this problem by identifying the species and viewpoint combination with the fewest examples and sets a maximum number of examples for all categories in a given epoch as a fixed multiplier of that minimum (the experiments here set the multiplier to 4).

As seen in Figure 3.13, the species-specific ROC curves achieve at least 96.7% AUC across all species in the WILD dataset. The species ROC operating curves in this figure are calculated by taking an average over the associated ROC curves for its respective viewpoints. Furthermore, the effect of species and viewpoint classification can be visualized in Figure 3.14. The overall accuracy of species and viewpoint combination classifications is 61.7% over 42 distinct categories for species and viewpoints combined. The accuracy improves from this baseline when we consider how slight changes in viewpoint impacts identification (i.e., a $\pm$ 45% degree shift in yaw is tolerable [261] for giraffes and zebras), which achieves an 87.1% "fuzzy" accuracy. The white squares in Figure 3.14 indicate different species, and any values in the matrix outside of the squares represent incorrect

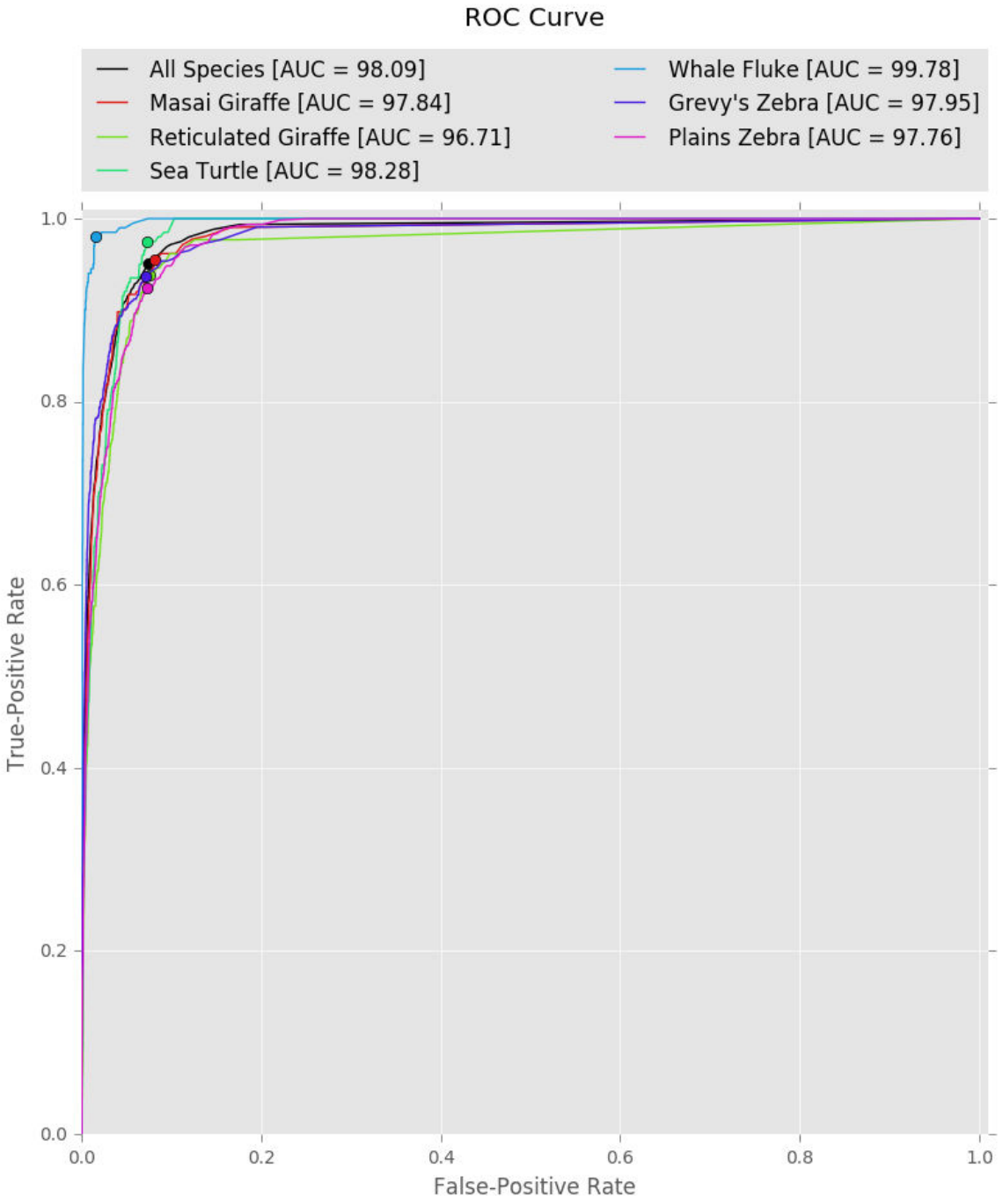


Figure 3.13: The ROC performance curves for the annotation classifier (labeler) suggests that the component is very accurate at predicting the species of an annotation. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

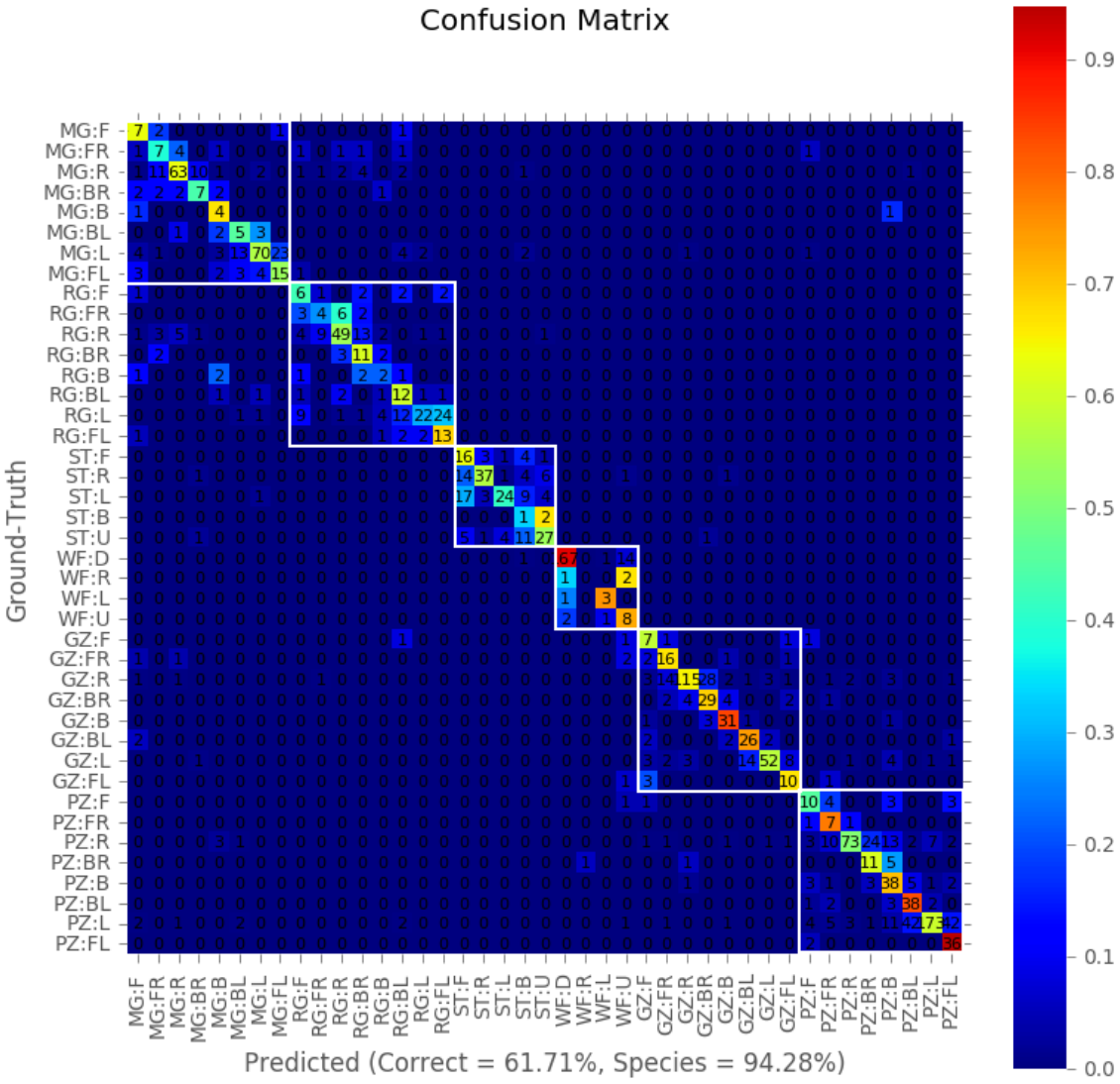


Figure 3.14: **The classification confusion matrix for the annotation classifier (labeler), marked with abbreviated {`species:viewpoint`}. The species abbreviations are: `MG` for Masai giraffe, `RG` for reticulated giraffe, `ST` for sea turtle, `WF` for whale fluke, `GZ` for Grévy's zebra, and `PZ` for plains zebra. The viewpoint abbreviations are: `left` (L), `front-left` (FL), `front` (F), `front-right` (FR), `right` (R), `back-right` (BR), `back` (B), and `back-left` (BL). The white boxes represent the separate species classes where values outside of these boxes indicate incorrect species predictions. The classifier predicted the correct species and viewpoint for 61.7% of the examples and the correct species for 94.3% of the examples. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.**

Figure 3.15: The locations of SIFT keypoints are not semantically constrained to the animal body and, if added to an identification search database, can confuse the ranking algorithm and exacerbate known issues like scenery matching. The coarse background segmentation model allows for SIFT keypoints to be down-weighted based on how much they contain background information. Yellow keypoints have a higher weight compared to blue keypoints.

species classifications. If we examine the errors outside the white boxes, then the inter-species classification accuracy is 94.3%. We can see that the majority of the inter-species classification error is between the two sub-genus species of giraffes (37.4% of the species classification errors) and some extra error between the two zebra classes (25.2%). These failures make intuitive sense as the species look relatively similar and can have subtle differences between their appearances at oblique viewpoints. It is worth noting here that the whale fluke and sea turtle species have almost no inter-species confusion, supported by their ROC AUC values of 99.8% and 98.3%, respectively. Of the species errors made, 62.9% are due to incorrect sub-genus giraffe and zebra classifications. In summary, the overall genus (zebras vs. giraffes vs. whale flukes vs. sea turtles) classification accuracy is 97.9%.

Now that we have bounding boxes with species and viewpoint labels, we need to consider how well the boxes represent the animals they surround. The use of axis-aligned bounding boxes is a good tool for finding animals but can be an inefficient structure when used to represent animals that are not rectangular. For example, giraffes have long legs and necks, and a rectangular bounding box around an animal can include considerable amounts of distracting background information. The next component is designed to address this problem by roughly distinguishing an animal from its surroundings.

3.5 Coarse Background Segmentation

The fourth detection pipeline component attempts to produce a coarse segmentation of an animal. The modifier "coarse" is intentionally added to the method description because it is not meant to be (or compete with) a pixel-level semantic segmentation algorithm [162], [190], [224], [365], [367]. Instead, a classification technique is presented that *approximates* a pixel-level segmentation by creating a binary classification map. The task is to take an annotation (with a species provided by the labeler) and roughly classify which pixels belong to that species versus the background. The goal is to generate a rough background mask that can eliminate or otherwise down-weight distracting non-animal pixel information. For example, the output of this component can be used to calculate weights for SIFT [39] keypoints and the features used by an identification pipeline. An example of this type of keypoint weighting can be seen in Figure 3.15 with the HotSpotter algorithm [261]. Key design features of this component are: 1) it only requires species-labeled bounding boxes for ground-truth and does not require fully-segmented images, 2) it is trained on small positive or negative patches as a binary classifier, and 3) it is applied fully convolutionally across an entire input image to produce a rough semantic segmentation map.

The annotation background segmentation approach uses a distinct type of neural network architecture called a Fully Convolutional Neural Network (FCNN) [27]. An FCNN is a special kind of CNN with no dense (fully-connected) layers and supports arbitrarily large input image sizes. This input size flexibility requires that the network be composed entirely of convolutions, pooling layers, or other non-rigid layers. This design feature is exploited by training the network on a fixed input size and performing forward inference on an arbitrarily large image (that must be equal to or larger than the training size). During training, 48×48-pixel input patches are reduced via convolutional and max-pooling layers to a 1×1-pixel patch with 128 channels (a spatial size of one pixel). It is then classified with a dropout layer($p = 0.4$) and a (1×1 Network-in-Network [49] convolutional layer with two outputs (binary classification). During inference, the network's output is expected to increase to W×H×128, where W and H are down-sampled resolutions of the original input size (at least 48 pixels), and the classification output is run to produce a binary classification map of size W×H×2. An FCNN can be efficiently applied across an entire image, without the need to resort to computationally intensive methods like sliding windows [18], shift-and-stitch [20], or memoization [368]–[370]. Importantly, the last layer's softmax activation is applied along the channel dimension, which means it can dynamically expand to the spatial output of a test image to create an automatic classification map.

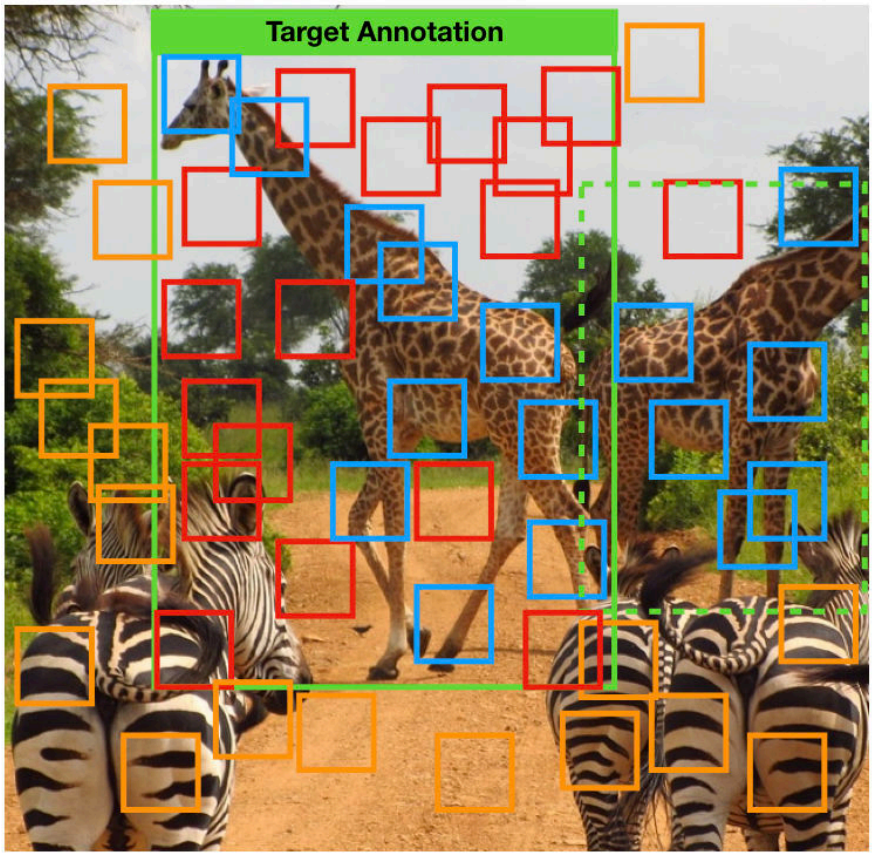


Figure 3.16: An illustration of the background segmentation patch sampling (using giraffes) and the utility of a cleaning procedure. The target giraffe (green, solid) has a collection of labeled positive patches (blue and red) and negative patches (orange) that are sampled outside the bounding box. The blue patches are *true* positives whereas the red patches are incorrectly-labeled *true* negatives. The goal of the cleaning procedure is to convert all red boxes into orange boxes automatically. Best viewed in color. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

3.5.1 Patch-based Training

The patch training data is generated by selecting a target annotation and resampling its image such that the bounding box has a fixed width of 300 pixels. Then, random patch locations are sampled uniformly across the image (and for a range of scales) where positive patches are centered inside the annotation (or an annotation of the same species), and negative patches are centered outside all annotations for that species. Positive patch exemplars are therefore species-specific and are meant to cover sub-regions within an animal body. In contrast, negative patches represent

background foliage, terrain, and other animals of different species.

The proposed positive patch sampling scheme can be problematic, however. The bounding box localizations of giraffes, for example, generally have large amounts of negative space around the neck and the legs (see Figure 3.16). When positive patches are sampled from inside giraffe bounding boxes, some are incorrectly labeled as positive that contain only negative background pixel information (red boxes). A self-supervised cleaning procedure is used during training to help correct label noise in the dataset. At the start of training, the network is given the original labels and asked to perform binary classification on the data as-is. Each time the learning rate decreases (and only after the model achieves an overall accuracy $\geq 90\%$), the currently learned model is run on the training and validation data to find any incorrect labels. Any label with a $\geq 95\%$ prediction of belonging to the opposite ground-truth label is automatically "cleaned" and its binary label flipped. The cleaning procedure has been found to help smooth out training and drastically improve the final results' qualitative performance.

3.5.2 Results with Fully-Convolutional Inference

Since the annotation background network was trained on noisy, patch-based data – and with the lack of fully-segmented ground-truth in WILD – a quantitative segmentation metric for the model's performance cannot be provided. However, looking at Figure 3.17, the background segmentation network performs well on various annotations of a known species to classify regions of the image as background and foreground. In this figure, the binary output masks of the background classification network are combined with their associated input annotations. Something to note is that the lack of distinction between class instances and animals with the same species in the annotation will not be masked out. Work by Crall (see Section 3.5.4 in [13]) shows the positive impact of using the coarse background segmentation; overall, identification matching accuracy improves when a background mask is used for feature weighting. The experiment presented in that work shows top-1 ranking improvements for Grévy's and plains zebra of approximately 5% for comprehensive ID experiments.

The creation of segmentation maps helps reduce the distracting background information within an annotation. However, we have not addressed the problem of knowing which annotations are producing distracting *foreground* information. For example, an annotation may show an occluded, blurry, or truncated (cut in half by a tree) animal and should ideally not be provided to ID. Therefore, the next pipeline stage takes a step back and determines which annotations in an image were likely

Figure 3.17: A grid of background classifications for six species shows that the component is able to learn useful background subtraction masks. These masks function as semantic segmentations between the species of interest and the background and do not distinguish animal instances. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

captured by accident. The general assumption is that incidental background sightings most likely have less useful visual information for ID and should be filtered out proactively.

3.6 Annotation of Interest (AoI)

The fifth and final detection component focuses on determining which animal annotations in an image are good candidates for identification. The goal of AoI classification is to try and answer the question, "*why did the photographer take this picture?*" and is tasked with distinguishing the animals that were the intended subject(s) (i.e., the "Annotations of Interest") from incidental background sightings of animals. It is important to note that this task tries to understand the composition of an image *a posteriori* to help guide ID and is not concerned with the aesthetic form of a particular image or calculated focus points. Instead, the Annotation of Interest (AoI) classifier identifies the most prominent animals in the image because they are likely to be the most identifiable. While state-of-the-art object detection algorithms are often compared and evaluated on their ability to localize *all* objects of interest captured in an image – regardless of the pose, lighting, aspect ratio, focus, scale, level of obscurity, or degree of truncation – a different objective is needed when animal ID is the intended use case: one that can be optimized for only detecting *identifiable* animals. To do this, though, we first need to know which annotations are even identifiable because the value of an animal detection should be fundamentally tied to the amount of identifying visual information it provides.

Figure 3.18 shows a motivating example for why the concept of AoI is needed. There are 22 plains zebras in this picture, presenting five different viewpoints and varying degrees of truncated and obscured animals. The classic formulation of object detection would expect 22 bounding boxes as the optimal output. In terms of identifiability, it is clear that the green highlighted box offers the best visual information out of all of the animals that are seen. The animal is well lit, in focus, not obscured or truncated, captured at a good resolution, and was perhaps the primary subject of the image when the photographer took the image. The dark blue box arguably ranks as the second-best zebra annotation but is slightly out of focus and is half-occluded. Did the photographer intend to photograph the blue animal, or was it simply in the background? Perhaps either way. The red box (upper right corner), in contrast, is an animal that is significantly obscured by grass and is most certainly an accidental capture because it offers almost no usable information for ID. While the red box may be a challenging detection to predict correctly (i.e., an understandable failure), we can expect that the light blue box to be within the capabilities of a modern, deep learning-based object

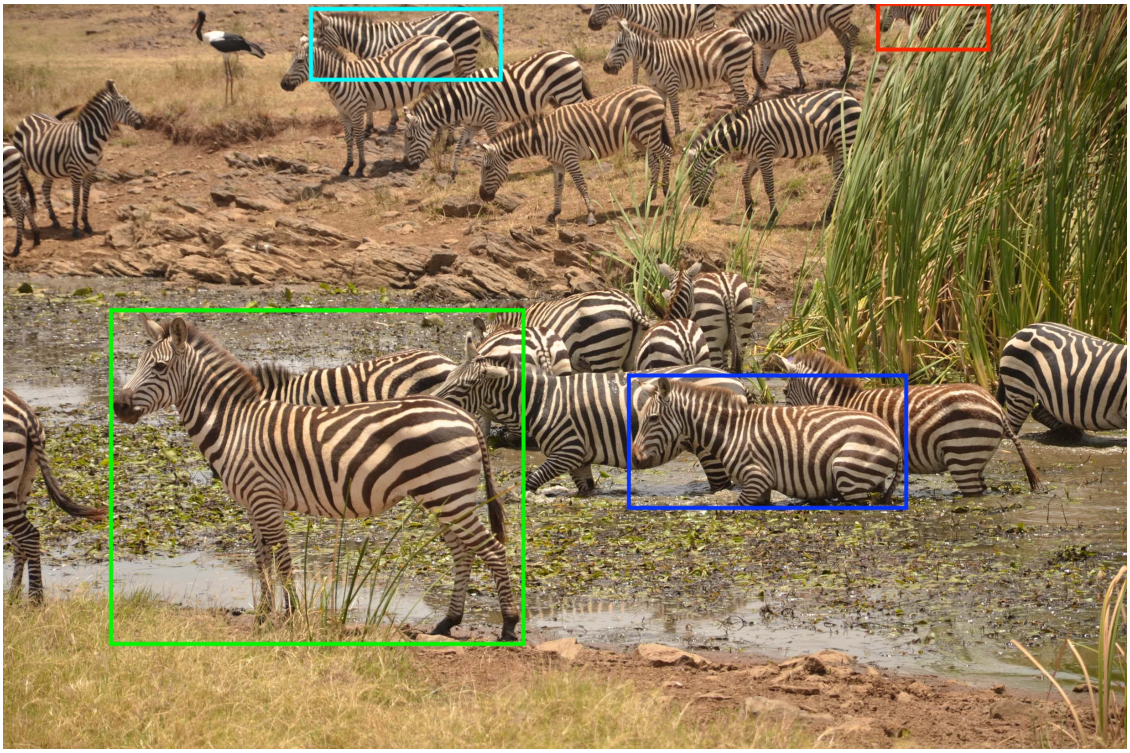


Figure 3.18: An example image captured by a citizen scientist during an animal photographic census. We may ask, "*which animal was the intended subject of this image (if any)?*" The animal with a green box around it is the Annotation of Interest for this image and all other animals should be considered incidental background sightings.

detector to find correctly (and predict a species and viewpoint label as a left-side plains zebra). The problem is that the light blue box could be considered a legitimate detection even though it offers little identifying information and is only slightly more valuable to ID than the red box. As such, the green box was most likely the intended subject considering the scene's composition and should be regarded as an AoI. Only the green box should be considered an AoI in this image, and all other animals are incidental sightings of animals in the background. The critical insight with AoI is that the localizer can de-prioritize failures for background animals because they often do not contribute meaningful ID information. To define the concept more formally, an AoI should have most or all of the following properties:

- is of a distinguishable individual animal (i.e., free-standing, well-lit, clearly visible),
- is relatively large and has decent resolution,
- is commonly located near the center of the image, and
- is in focus and not blurred

Conversely, an annotation should not be considered an AoI if it has one or more of the following opposite properties:

- is a part of an overlapping herd or group of animals
- is relatively small or contains few pixels
- is out of focus or is otherwise blurry
- is located around the edges of the image
- is occluded by other animals or objects by area $\geq 25\%$
- is off the edge of the frame of the image by area $\geq 25\%$

The properties of AoIs demand that an annotation not be reviewed in isolation (i.e., by only viewing its cropped sub-region). The decision that an annotation is an AoI must be made by weighing the entire image context as well as against any other accompanying annotations. This process is naturally subjective and can be hard to determine reliably for borderline cases. Further, because these conditions are relatively strict, there are rarely more than one or two AoIs in a particular image, and some images with detected annotations have no AoIs. In summary, the reason to use Annotations of Interest is motivated by its two primary use cases:

1. preventing partial-animal, background, and otherwise visually distracting detections from entering an automated animal identification pipeline, and
2. training citizen scientist volunteers on how best to take high-quality images for photographic censusing.

Annotation of Interest was the first attempt at addressing the annotation filtering problem. Subsequent analysis in Chapter 5 describes, evaluates, and deploys a more thorough and successful approach to the annotation filtering problem for animal ID. Nevertheless, the following discussion is still helpful since AoI is an image-level determination for the subject of an image and still helps with accurately tuning the localizer for finding well-formed annotations.

3.6.1 AoI Ground-Truth & Labeling Variability

The decision to label an annotation an Annotation of Interest is inherently subjective. It is, therefore, important that an image and all of its annotations be analyzed by multiple reviewers when ground-truth labels are being generated for training and evaluating AoI methods. The ground-truth annotations in the WILD dataset were distributed to five different teams for labeling. Each team had different background training and skillsets and – even when provided with the exact same definition of an AoI and a handful of examples – had different implicit reasonings and motivations for their

Table 3.4: The number of the ground-truth AoI decisions made by different teams of human reviewers. The teams were given the same instructions but split by their respective domains of expertise.

Team	Group	AoIs
1	Ecologists	5,166
2	Data Engineers	6,829
3	Computer Vision	5,729
4	Data Scientists	6,174
5	Author	5,547

decisions. A brief description of each team is shown in Table 3.4 along with the total number of AoIs they marked (out of 9,871 considered annotations in the dataset). The final ground-truth AoI labels for the annotations in WILD were created with a simple majority rule, requiring at least three out of the five teams to agree that an annotation needed to be considered an AoI.

The distribution of the labeling work was parallelized in two ways: across teams to increase redundancy and within teams to increase throughput. The "ecologist" team was comprised of biologists and *equid* experts, plus an array of undergraduate and graduate students, at Princeton University. Working alongside the team of ecology researchers, a "data engineering" team comprised of data and software engineers based in Portland, OR also reviewed the same images to make independent AoI decisions. The third "computer vision" team included computer vision and algorithm researchers at RPI, and the "data scientist" team was a group of data scientists and social network researchers at the University of Illinois, Chicago. Each image in WILD was reviewed by at least one ecologist, one software engineer, one computer vision researcher, and one data scientist. Lastly, the author of this dissertation worked alone (as the creator of the AoI concept) to independently label all of the annotations in the WILD dataset. The variance between the various teams is relatively high, with the ecologist team deciding only 52.3% of the annotations were AoIs compared to 69.2% by the data engineer team. The average number of AoIs is 5,889$\pm$571 across all teams, representing 59.7% of all detections in the dataset, but only 3,598 annotations (36.5%) had a majority vote by at least three teams.

With ground-truth labels for AoI assigned to the dataset, we can analyze the unique qualities of AoIs and how they present themselves in images. Figure 3.19 plots the spatial distributions (on a normalized unit square) of AoI bounding boxes (right) as compared to all annotation bounding boxes (left) in the WILD dataset. We can see that the centers of AoI bounding boxes seem to be

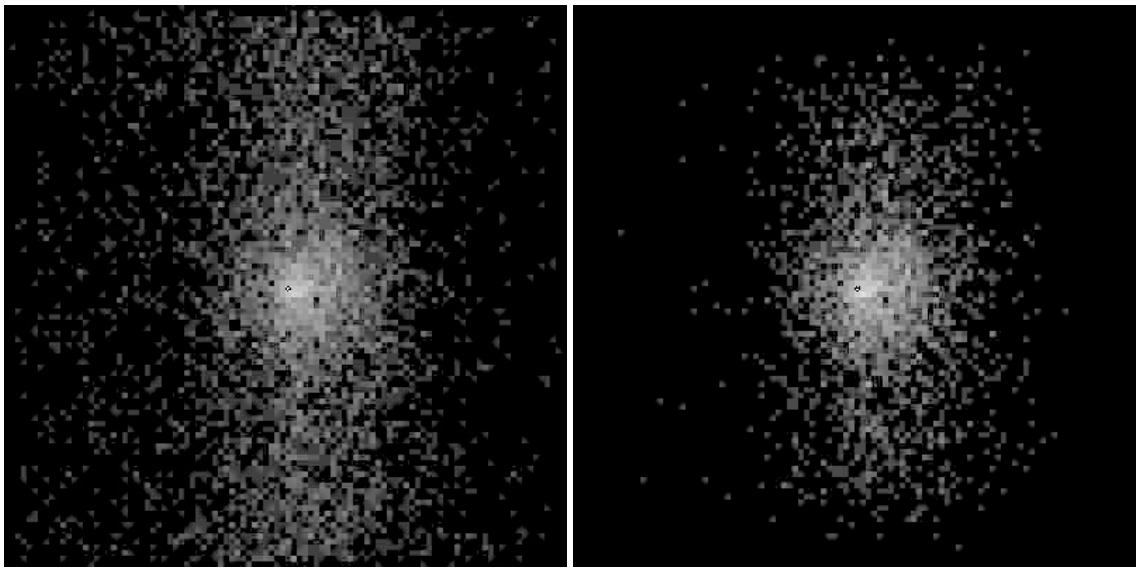


Figure 3.19: The distribution of bounding box center locations (on a unit square) for all annotations (left) and AoIs (right). Annotations of Interest are much more uniform and biased towards the center.

biased towards the center of the image. The area of AoI bounding boxes (shown in Figure 3.20), as calculated by a percentage of the area of the whole image, also shows a clear inverse correlation with annotation size. These results make intuitive sense because 1) an AoI is most likely going to be near the center of the image because it is strongly associated with finding the subject of an image and 2) any annotation that occupies under 20% of the image area is most likely not an AoI because it is not sufficiently large to capture enough visual detail for the animal.

3.6.2 Results

The AoI classifier has a very similar convolutional and dense layer structure to the whole-image classifier (WIC) component, except for three differences: 1) it takes as input a 4-channel input image, comprised of red, blue, and green color channels stacked with a fourth annotation bounding box mask, 2) the output layer (with a softmax activation function) has only two outputs for simple binary classification, and 3) the network weights are optimized using categorical cross-entropy loss. Examples of positive and negative training input images can be viewed in Figure 3.21. The end goal of the AoI classifier is to eliminate the need to perform identification processing on the background and partially visible animals, and we will see in later chapters causes an increase in the total amount of work needed by human reviewers.

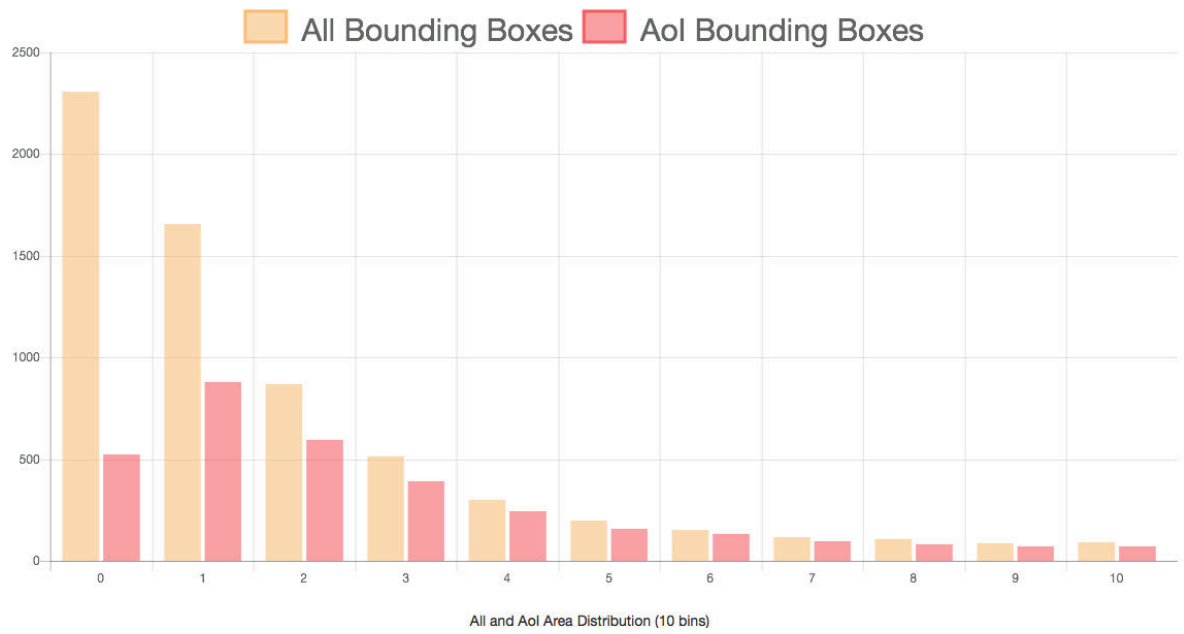


Figure 3.20: A histogram of the total number of annotations and AoIs (y-axis) as a function of the percentage of the image area (x-axis, in 11 buckets each with a size of 10%). This shows that AoIs, compared to annotations in general, are much less likely to be small annotations.

The AoI classifier achieves an overall accuracy of 72.8% on the held-out test data (521 true positives, 1,268 true negatives, 506 false positives, 164 false negatives) when using a confidence threshold of 84%. Figure 3.22 shows ROC curves for each species. Ironically, the AoI classification performance of plains zebras shines, further supporting the claim that the background annotations for plains zebras in WILD are not good identification candidates. However, we can see that the sea turtle and whale fluke AoI results are very close to random. This poor performance is not surprising because those categories have the lowest percentage of AoIs relative to the number of annotations for that class, and the classifier struggles on solitary species with ambiguous AoI definitions. The AoI classification is objectively the worst-performing component of the detection pipeline as it struggles with the overall ambiguity of the concept. However, the primary goal of AoI selection is to reduce the overall number of poor annotations that are passed along to an identification pipeline. From this point-of-view, the AoI classifier correctly eliminates from processing 71.5% of background annotations at the cost of missing 23.9% of the positive AoIs. Furthermore, the ground-truth AoI data helps to better configure the localizer models during test time.

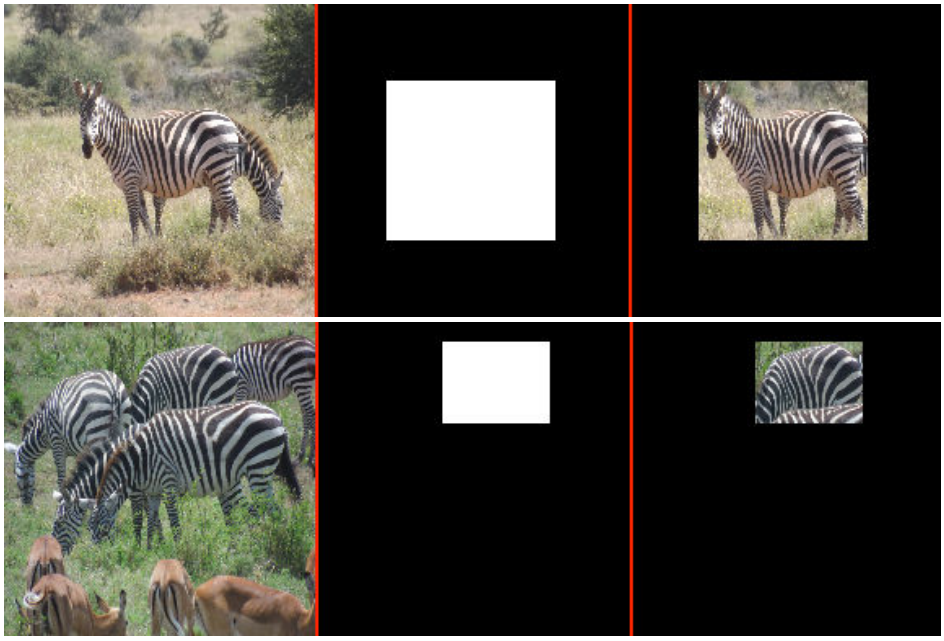


Figure 3.21: A positive AoI training example (top row) is comprised of the resampled RGB image (left) and the annotation segmentation mask (middle). The right-most column depicts their combined representation. As shown in the negative example (bottom row), the masked annotation is of an occluded, background animal and is not an AoI. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

Let us now review the entire detection pipeline. The detector pipeline produces a set of bounding box annotations from pre-filtered images. Each of those annotations has a species and viewpoint label, a background mask, and an AoI classification, which can all be used to filter out irrelevant or distracting annotations or reduce distracting scenery. Furthermore, the design of each of the above components puts few requirements on training data and can be bootstrapped quickly for new species using only bounding boxes. While the specific components in this pipeline are not novel implementations – and an in-depth analysis using alternative methods is not provided for most of the components – that is not the core contribution. Instead, the point of the pipeline is to define a modularized structure for general animal detection, and its claim to novelty is in its comprehensive understanding of how animal detection is needed for real-world applications on animal ID.

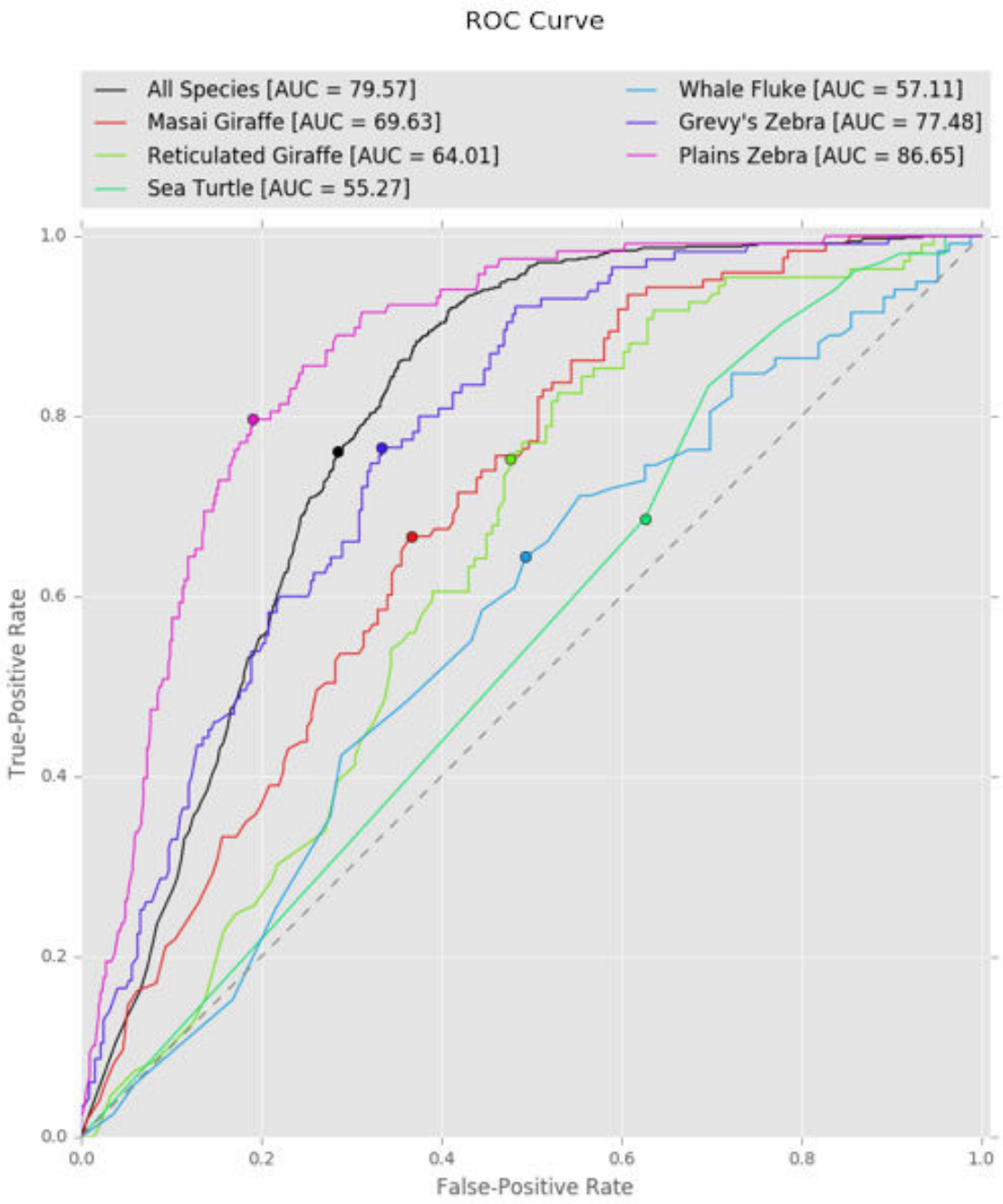


Figure 3.22: The ROC performance curves for the AoI classifier. The species with the best AoI classification performance is plains zebra, mostly due to the lower AoI to annotations ratio. The AoI classifier performs the worst on whale flukes and sea turtles because it is harder to tell when solitary animals should be considered AoIs. ©2018 IEEE. Reprinted, with permission, from: J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

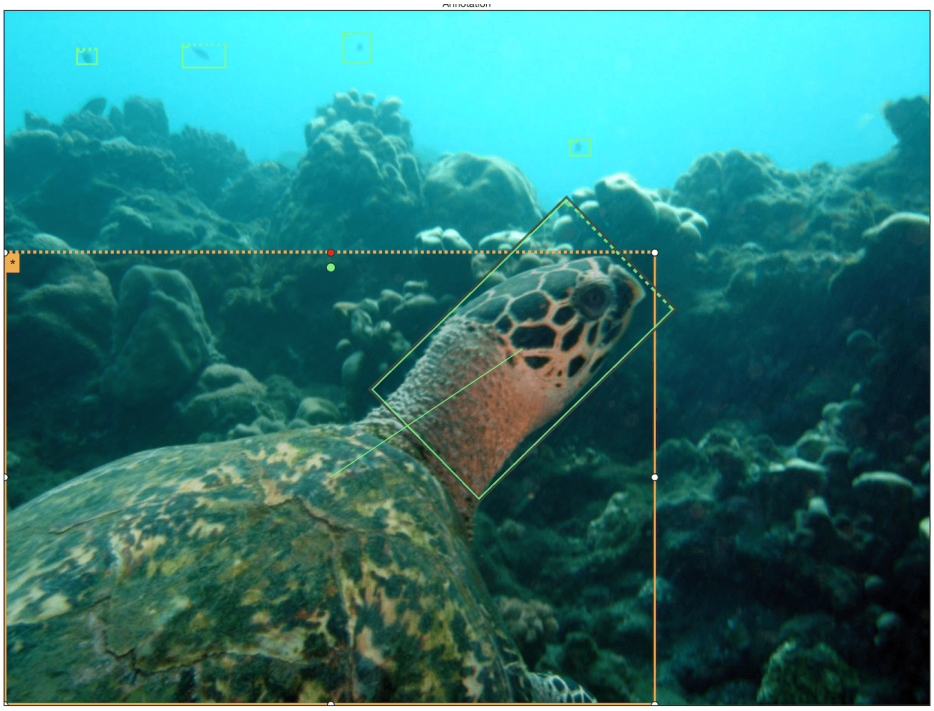


Figure 3.23: An example annotation of a sea turtle (orange) with a rotated part annotation for its head (green). The dashed line indicates the "top" of the annotation.

3.7 Additional Components & Applications

The core detection pipeline presented up till this point can be extended to work with additional components and specific application use cases, which we will explore in this section. All of the components presented here should be considered optional; these examples are also intentionally separated from the above pipeline because they require additional ground-truth training data and should be implemented as needed to keep the burden low on human reviewers.

3.7.1 Annotation Bounding Box Orientation

One of the most significant optional components has the goal of rotating axis-aligned annotations produced by the localizer[17]. The reason orientation is essential is that ID algorithms can be susceptible to rotation and, for example, can fail to match an annotation correctly if it is upside down compared to a database of consistently rotated annotations. The orientation of annotations

[17]Portions of the work described in this section were completed by Olga Moskvyak [263] under an unpublished research contract with Wild Me, a Portland, OR not-for-profit. The author of this dissertation designed the orientation component, its mathematical definition, and evaluated its impact on ID results, but the external collaborator did the orientation implementation and a stand-alone performance evaluation. All work and results are reproduced with permission.

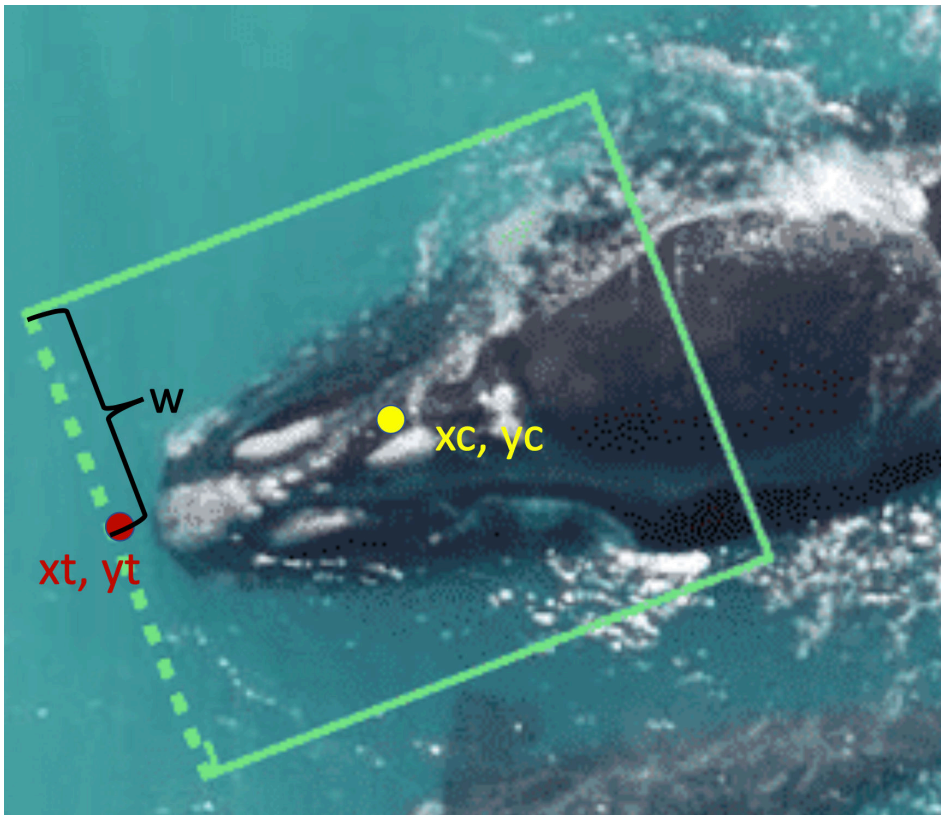


Figure 3.24: The regression network is designed to predict five values: x_c, y_c (in yellow) give the center of the new bounding box, x_t, y_t (in red) give the center the "top side" of the box, and the width w of the bounding box.

should not be confused with its viewpoint. The viewpoint is the side of the animal that is seen (e.g., left-side), but orientation is the rotation on the annotation such that it has a normalized view (e.g., putting the feet of a left-side zebra at the bottom of the annotation). Furthermore, if we rotate an axis-aligned bounding box that is snugly fit to the animal, the new box may be slightly too large or too small in a given direction. Therefore, we want to rotate the original bounding box and modify it to fit appropriately. An example rotated ground-truth annotation can be seen in Figure 3.23 for a sea turtle head used for ID.

The input to the orientation component is an image with an original axis-aligned rectangle bounding box from the detection pipeline. The component's output is an oriented and directed bounding box specified by a new bounding box (x-axis top-left, y-axis top-left, width, height) and rotation angle (θ) applied at the center of the box. The network's architecture was designed to be used with multiple species and has a generalized training procedure. The network was trained on sea turtle heads, sea dragon heads, right whale bonnets from an aerial viewpoint, hammerhead

Table 3.5: The performance accuracies for the orientation component. The predicted orientations are correct within 20 degrees for the majority of species.

Species	±10 Degrees	±15 Degrees	±20 Degrees
Sea Dragon Heads	95.20%	97.73%	98.11%
Whale Shark	87.91%	93.28%	94.63%
Sea Turtle Heads	84.64%	91.64%	94.71%
Spotted Dolphin	81.04%	88.08%	91.83%
Right Whale	81.34%	83.92%	84.78%
Manta Ray	67.55%	74.96%	79.28%
Hammerhead Shark	52.19%	61.56%	66.14%

shark bodies, manta ray bodies, and spotted dolphin bodies. The component is implemented as a regression network and produces five floating-point values x_c, y_c, x_t, y_t, and w. These values are illustrated in Figure 3.24 for a right whale head. In particular, x_c, y_c (in yellow) gives the center of the new bounding box, x_t, y_t (in red) gives the center the "top side" (indicated by a dashed line) of the bounding box, and the width w of the bounding box. The height of the rectangle is defined as $2 * sqrt((x_c - x_t)^2 + (y_c - y_t)^2)$ and there is no constraint that the height must be greater than the w width value. The direction from the center point to the center of the top side is calculated as $atan2(y_t - y_c, x_t - x_c)$ and the range can cover a full 360 degrees. If w', x'_c, y'_c, x'_t and y'_t are the outputs for a given ground-truth box, then the loss is defined simply as:

$$L(w, x_c, y_c, x_t, y_t) = (w - w')^2 + (x_c - x'_c)^2 + (y_c - y'_c)^2 + (x_t - x'_t)^2 + (y_t - y'_t)^2 \qquad (3.3)$$

Orientation models were trained using the same model architecture and the same training setup for all species. Extensive data augmentation was used to extract a randomly rotated annotation for each mini-batch. The performance of the component varies from species to species: good results were achieved on sea dragon heads, whale sharks, sea turtle heads, spotted dolphins, and right whales where ground truth annotations are consistent, but predicting orientation on manta rays and hammerheads was challenging due to variety of underwater viewpoints and poses. The accuracy of predicting an angle of orientation on a test set at 10, 15, and 20 degrees thresholds can be seen in Table 3.5.

Lastly, we can compare the impact of orientation on ID performance. Orientation results for

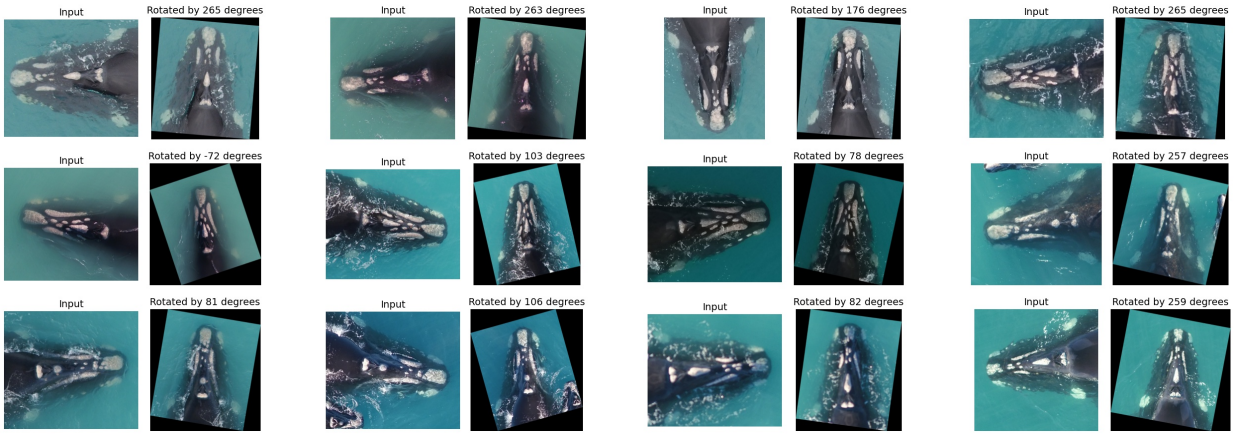


Figure 3.25: The head of a right whale has white callosity patterns that can be used for ID. Orienting the head detections to point up improves ID performance.

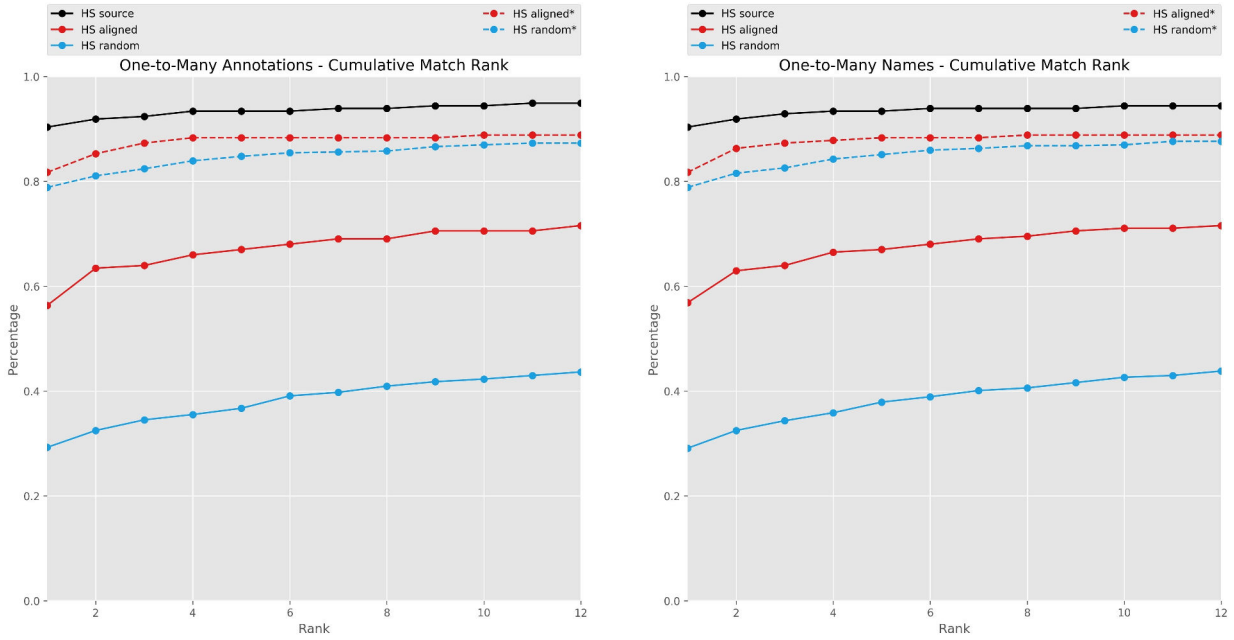


Figure 3.26: The top-k recall performance curves for the HotSpotter algorithm on right whale bonnets. The ground-truth annotations (black line) shows stellar ID performance while randomly rotating the annotations (blue solid) and axis-aligned boxes (red solid) show significantly worse performance. Using the orientation network to rotate the random boxes (blue dashed) and the aligned boxes (red dashed) significantly reduces the recall error and approximates the ID performance of hand-drawn boxes.

held-out right whale images are shown in Figure 3.25. The benefit of having oriented annotations (right for each column) is that the annotations are much more directly comparable, allowing an identification algorithm to match the images without handling orientation by itself. Using the HotSpotter ID algorithm [261] to match the right whale images, Figure 3.26 shows that orientation plays a significant role in accurate ranking performance. We can see that ground-truth box ID performance (black line) is very good at approximately 90% top-1, and randomly rotated boxes that are then fixed by the orientation network (blue dashed line) are only about 10% worse in match performance. Furthermore, the corrected boxes (red dashed line) perform significantly better than the original axis-aligned boxes (solid red line) produced by the detector, improving ID accuracy by over 20% rank-1.

3.7.2 Part Bounding Box Localization & Assignment

A second use case that is very useful is localizing specific parts for an animal[18]. A part of an animal is sometimes the better candidate for ID compared to the full-body annotation. For example, a sea turtle – as discussed in Chapter 2 – does not have reliable ID information on the shell. The patterns change over time, and more stable facial patterns can be used for visual ID. The localizer component in the detection pipeline is optimized for finding the complete body annotation for a given animal. However, often a part like a head or an ear needs to be localized as well. We can see an example of this from the orientation network discussion and Figure 3.23 where a part bounding box for a sea turtle head is added to an existing annotation for the sea turtle's body. Using parts is also convenient for some ID algorithms, like CurvRank [291] that rely on contours. For example, extracting part bounding boxes for dorsal fins or elephant ears can be beneficial since contour-based algorithms need a consistent starting point and a way to find specific parts of an animal. The interface shown in Figure 3.27 gives an example of elephant ear detection that was produced by the detection pipeline that was trained to find left-side only ears for African elephants (*Loxodonta*).

The detection of parts alongside body annotations presents two unique challenges for the detection pipeline. First, the localizer's standard non-maximum suppression (NMS) technique must be context-aware, ensuring that body annotations do not suppress part boxes that will likely significantly overlap. Second, part boxes need to be associated with a parent body annotation

[18]The work on the part-body assignment component was done by Drew Blount, an employee of Wild Me, and it is included here for completeness. All work and results are reproduced with permission.

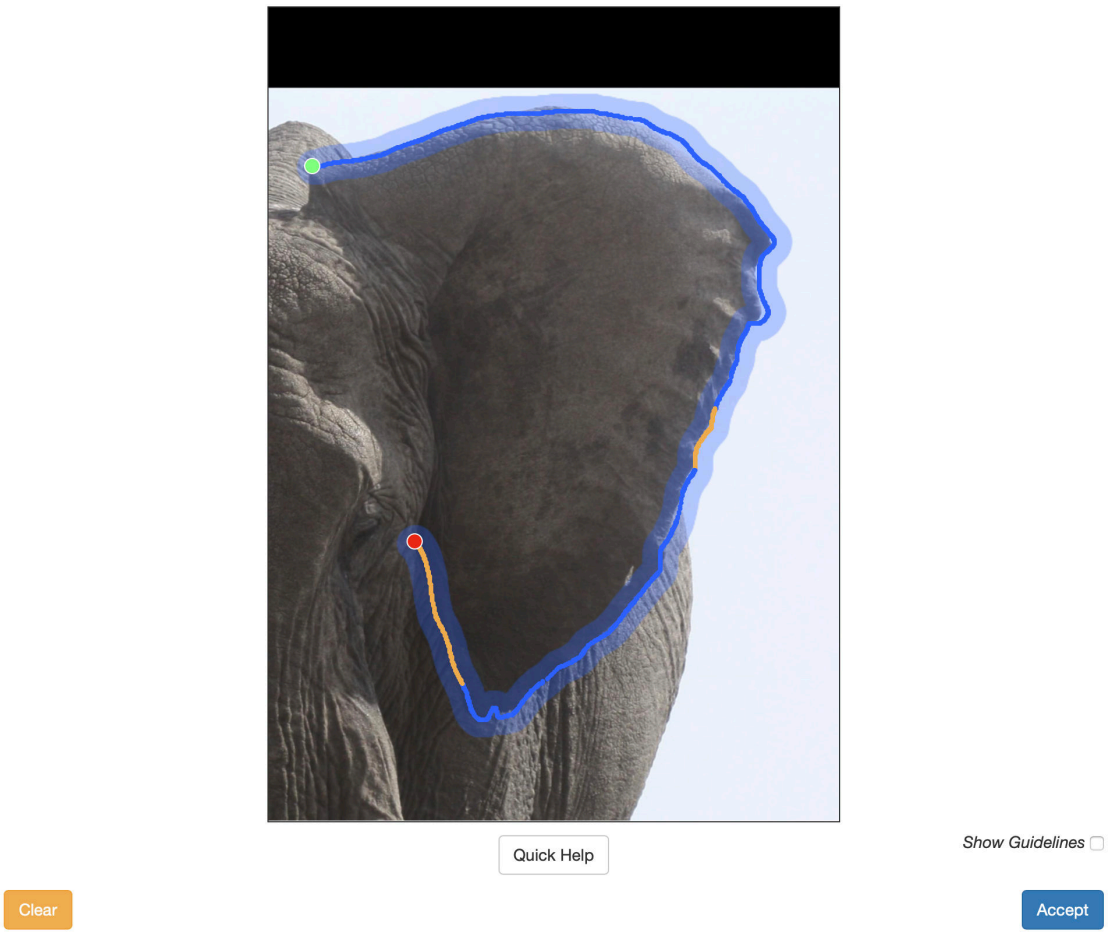


Figure 3.27: An example outline contour (blue line) of an African elephant ear detection, with occluded regions (yellow) highlighted by a reviewer.

so that the ID results are associated correctly in the ID database. The NMS problem is solved by treating parts as distinct from body annotations and applying the traditional algorithm to both sets independently. Once NMS has been applied separately, the resulting boxes are combined to form the final predictions. The second problem of assigning the boxes was achieved by training a random forest classifier on a hand-engineered vector with 37 feature dimensions. The feature vector encodes the location of two annotations' bounding boxes, the center locations of their boxes, their respective areas, the distance between their centers, the amount of overlap, and other geometric values that are scaled to have are unit size. The classifier is then run on all combinations of predicted annotation and parts from the detection pipeline to produce a confidence value for how likely a given part should be matched with an annotation. A greedy assignment algorithm then attempts to assign the most confident parts assignments to an annotation until either all parts are assigned,

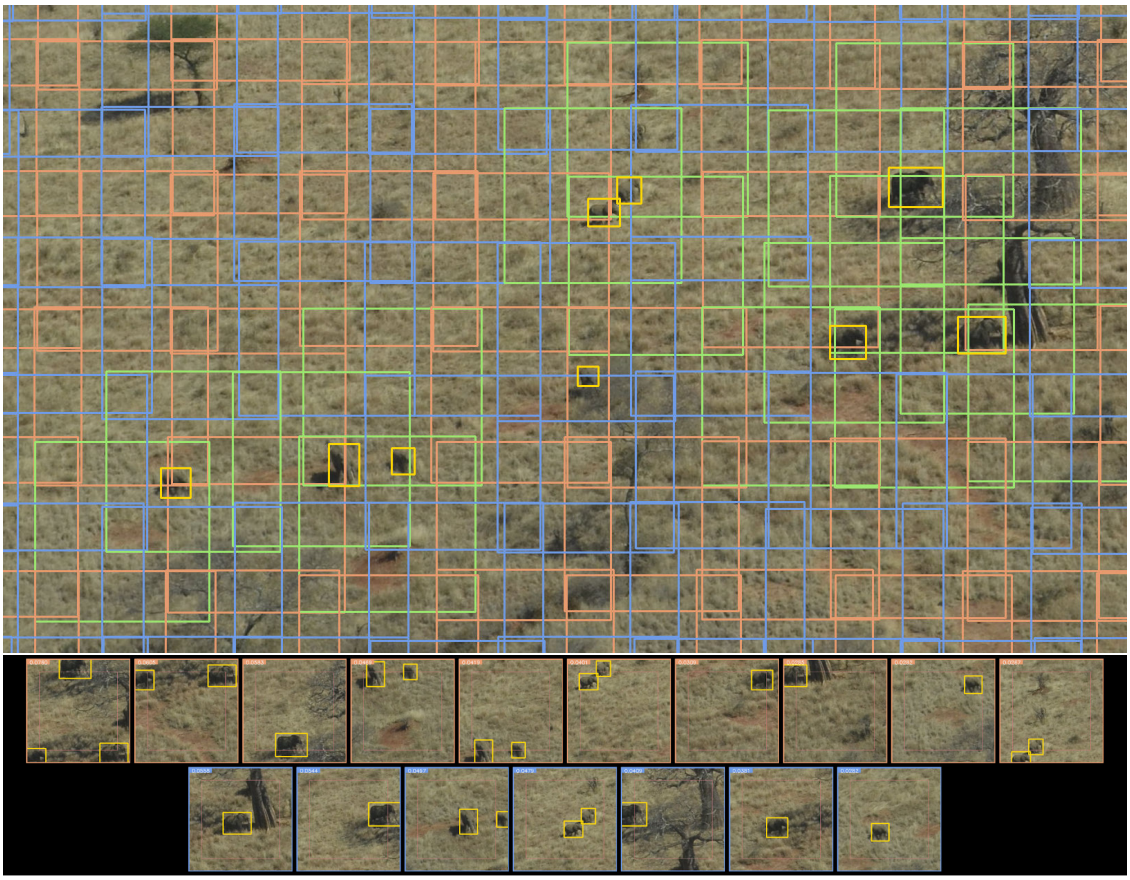


Figure 3.28: The detection pipeline can be run on tiles extracted from aerial imagery to find animals for population abundance surveys.

all annotations have precisely one part assignments, or a global minimum confidence threshold is reached. When evaluating sea turtles, this "assigner" component is 88% accurate for held-out test part-body assignments. Most of the images in a test set that had errors were from duplicate detections that could not be associated. For the images that contained multiple turtles and heads, however, 98% of the assignments were correct.

3.7.3 Image Tiling & Overhead Imagery

We next examine a different use case for the detection pipeline. The pipeline is designed and verified extensively in ground-based animal ID applications, but it can be modified slightly to process overhead aerial imagery for wide-area population counts [371]–[373]. Aerial surveys offer a unique challenge to the detection pipeline because 1) fast filtering of negative images is needed (the vast majority will not show any relevant activity or animals) and 2) positive images are hard to

find [374]. In addition, animals that are photographed at altitude can be challenging to detect due to being very small (sometimes only a handful of pixels across), occluded by foliage like tree cover, or are not very distinctive against the background terrain (e.g., the sleek, round back of an elephant can look like a grey boulder in some lighting conditions). Thus, to accurately apply the detection pipeline on aerial images, the analysis needs to be restricted to small areas of an input image so that the relative scales of the animals are more appropriate for the pipeline.

The detection pipeline uses a localization component (YOLO) that computes results from a 448×448-pixel image. If an original aerial image were down-sampled to that resolution, any animals in the photo would be reduced down to, optimistically, only a few pixels, and reliable detection would be impossible. The detection pipeline can be modified slightly to first extract a grid of overlapping tiles across the image. The existing components can then be applied to smaller regions of a more proper native resolution. Figure 3.28 shows a small section of an aerial image that was taken during an elephant population survey. The image was tiled up with two overlapping grids (an orange grid and a second blue grid) of smaller 500×500-pixel regions. The orange grid is overlaid onto the image to densely cover as much of the image area as possible while at the same time not creating partial tiles (respecting a margin on the border). Each adjacent tile in the orange grid overlaps by 25%. Animals on the margin of a tile are therefore analyzed multiple times. A second blue grid is extracted using the same process as before to prevent edge cases between tiles; this second grid has a global 50% shift and centers all of its tiles where the corners of orange tiles meet. An additional set of grey overlapping tiles (not pictured here) is also extracted along the border of the input image to capture any additional missing animals.

The tiles that contained ground-truth elephant bounding boxes (in yellow) are marked as positive tiles and are highlighted with a green border. For example, the figure has nine ground-truth elephant bounding boxes and shows ten orange and seven positive blue tiles that contain elephants. The complete set of positive tiles and their ground-truth detections can be seen at the bottom of Figure 3.28. The detection pipeline is then trained and applied like normal by treating tiles as its input: the whole-image classifier can be used to identify tiles that are likely to contain animals, the localizer finds bounding boxes within tiles for animals, and the labeler can be used for species classification. Since these animals are likely not captured at a sufficient resolution for ID, the motivating goal of the detection pipeline should be to maximize the accuracy of counting. Furthermore, the lack of detail suggests that the work to produce a coarse segmentation is not particularly worthwhile. Likewise, the AoI filtering is not applicable in these situations and can be

ignored. After the detection pipeline is applied on a series of tiles, the results are then re-mapped onto the original input image. Finally, non-maximum suppression is applied to the aggregated animal detections to eliminate duplicates and provide the final output.

3.8 Summary

The detection pipeline proposed in this chapter provides a comprehensive process for processing raw images of animals into sightings that are useful and relevant for identification. The pipeline uses a whole-image classifier to filter images quickly, a localizer and labeler to produce annotations with species and viewpoint labels, a coarse background classifier to produce approximated segmentation maps, and an AoI classifier to identify foreground vs. background annotations. Further, the pipeline can be extended to add functionality for orienting annotations, localizing and assigning parts of animals with detected bodies, and processing imagery from aerial surveys. The entire pipeline is designed to be easily bootstrapped for new species and relies on training data – bounding boxes with various metadata – that is quick to annotate and trivial to parallelize. Finally, the pipeline's contribution is evaluated on two new datasets and shows that it can produce high-quality candidate annotations for animal ID.

As a summarizing example, the output of the detection pipeline can be seen in Figure 3.29 as applied on the example image we started this chapter with (Figure 3.1). All of the annotations were detected as plains zebras by the annotation classifier. The WIC suggested the species in the image were Grévy's and plains zebras, with Plains zebras being the highest score (a correct classification). All other species (giraffes, whale flukes, sea turtles) had a score of less than $1e{-}4$, without making contextual assumptions about which animals would have been impossible based on GPS location (e.g., you would not find a whale fluke out in the African Savannah). The AoI classifier selected two annotations (in red) for further processing by the identification pipeline. Identifying only a predetermined side of the animals is critical because their visual appearances are not symmetric between left and right. As such, the only annotation that would be processed from this image would be the most foreground animal, facing left, and its corresponding annotation in the bottom left corner of the figure.

Overall, the detection pipeline is able to achieve a whole-image classification accuracy of 64.8% for 6 species but rarely (around 3.5%) produces false negatives, making it an ideal first-pass filter; the localization component has a mAP of 81.7% for 6 species but is able to perform much better on primary animals (Annotations of Interest) with a mAP of 90.6%, which makes

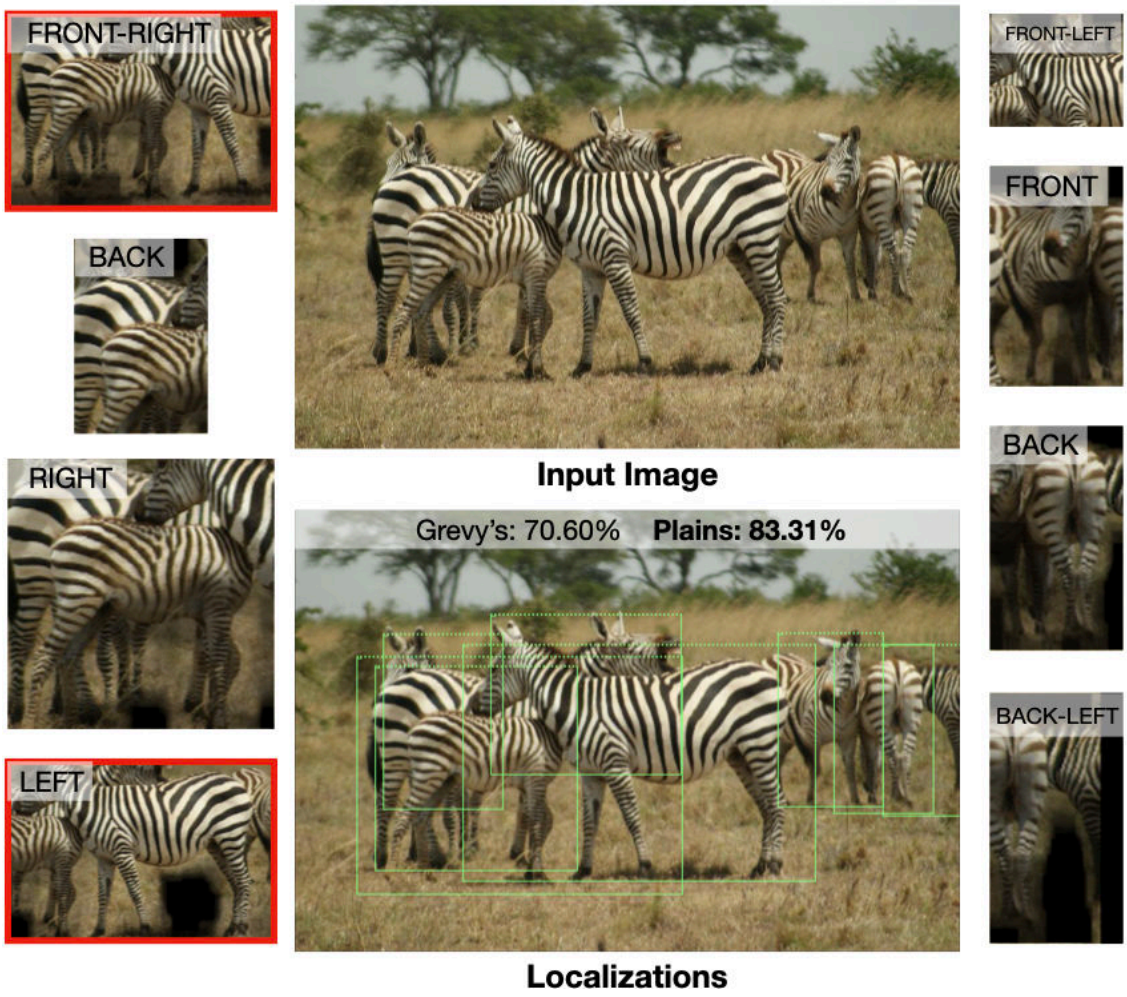


Figure 3.29: The output of the detection pipeline on Figure 3.1. The WIC produced a classification of 83% for plains zebra (and 71% for Grévy's), and the localizer found eight annotations. The labeler's output can be seen for each annotation box, and the AoIs are highlighted in red. For photographic censusing, picking the left-side plain zebra AoIs filters the output to only one annotation, the desired one for ID processing.

it a promising approach for finding detections that are useful for ID; the labeler has an accuracy of 61.7% over 42 unique categories but is able to accurately estimate approximated viewpoints 87.1% of the time and the correct species for 94.3% of examples, providing an accurate way to filter for relevant and comparable annotations; the coarse segmentation algorithm does an excellent job at providing background weights for matching algorithms while only being trained on bounding boxes, which has been shown to improve ID performance by around 5% for zebras; lastly, the AoI component has an accuracy of 72.8% across six animal species but the success of the concept

shows that it is possible to filter annotations based on how identifiable they are. The Wildlife Image and Localization Dataset (WILD) is also introduced, which contains 5,784 images and 12,007 labeled annotations across 30 classification species and a variety of challenging real-world detection scenarios; the DETECT dataset is also contributed for 6,655 annotations for plains and Grévy's zebra. The use cases of the detection pipeline have also been successfully demonstrated to correctly orient 87% of annotations (within 20%) for seven species, find and associate parts of animals to the body of animals, and detect animals in aerial surveys.

The detection pipeline, as a whole, is a significant contribution because it demonstrates that it is possible to automate wholesale amounts of the tedious image preparation work for photographic censusing. Of course, the detection pipeline is not perfect. The discussion above has shown that most of its mistakes are trivial or concern animals that fundamentally do not matter for ID. Even so, the pipeline's modular design allows for better components to be substituted over time to increase performance without needing to re-train or re-implement other algorithms. Finally, a strict focus on individual component accuracy misses the point that poor automation is a more significant barrier to large-scale population censusing than poor detection performance or completeness.

CHAPTER 4
OVERVIEW OF PHOTOGRAPHIC CENSUSING

Animal population monitoring is hard to do at large scales because it is immensely laborious for ecologists. It is often too overwhelming and tedious to track hundreds – let alone thousands – of animals with invasive tools like ear tagging and GPS collars. Furthermore, historical methods like aerial surveys and hand-based counts lack the ability to recognize individual animals over time, severely limiting the potential to establish ecological trends. Knowing an animal population through the majority of its members, in contrast, provides crucial insight. For example, ecologists gain a more intimate and timely understanding of a species' health when they can determine life expectancy, visualize migration patterns, and quickly measure the effectiveness of conservation policy. A comprehensive database of animal IDs is, therefore, a powerful tool for monitoring an endangered population, and the question is, "*what is the best way to build one?*" Unfortunately, a large database of IDs cannot be built by hand because the amount of work required to curate it is too prohibitive as the database grows. What is needed for sustainable population monitoring is automation. Any end-to-end solution to the problem of large-scale population monitoring must fundamentally be built around maximizing the completeness of an ID database while minimizing the amount of human effort needed to create it.

This chapter introduces the concept of *photographic censusing*[19], a comprehensive and automated procedure for large-scale animal population monitoring. The methodology uses digital images of animals as input, machine learning algorithms to automate their analysis, an ID database to track sightings of individual animals, and a management algorithm to control when human interaction is needed. Furthermore, photographic censusing is intended to be feasible for resource-strapped organizations to implement for large species and migratory ranges. As we will see, photographic censusing has been experimentally validated *in situ* on a wild animal population with thousands of individual members across 25,000 square kilometers. Photographic censusing is also intended to be used with open populations, with some animals only seen once (what we will call "singletons") and others seen many times ("multitons"). This chapter begins with an overview of photographic censusing and discusses real-world challenges and considerations when high degrees of automation are needed. Next, an enumerated list of the required machine learning

[19]As a process that is powered by computer vision algorithms, animal population censusing was first explored in the author's master's thesis [2]. It is formally proposed and described here as a self-contained and complete methodology.

and management algorithms is provided, and a mathematical framework is introduced for estimating the size of an animal population when automated components are involved. Lastly, a new evaluation database is proposed containing hundreds of animal IDs with multiple sightings, some over two years, for the endangered Grévy's zebra species in Kenya.

4.1 Problem Description

Photographic censusing produces a population estimate by performing a sight-resight study (see Section 2.3.2.1). A sight-resight study is based on the sampling statistics of capture-mark-recapture and, as a result, is constructed from an initial collection of photographs followed by a second, independent, and comparable collection. The image data needed for photographic censusing is gathered through a *photographic censusing rally*, an event where volunteer "citizen scientist" photographers are trained to take pictures of the desired species and tasked to cover its residential area for two back-to-back days. The question is, given a large collection of images taken during a two-day censusing rally, "*how many individuals are in the population?*" Photographic censusing answers this question by building a comprehensive database of animal IDs with the help of automated tools. The ID database is curated by comparing different pairs of sightings to determine which show the same animal or not. The process concludes when a management algorithm is satisfied with the level of redundancy and consistency for all IDs in the database. By comparing the ratio of animal IDs seen on day 1, seen on day 2, and seen on both days, the total number of animals in the population is estimated. This process can rely heavily on human decision-making, so the overall significance of photographic censusing as a solution is fundamentally tied to how automatically it can produce a reliable database of animal IDs.

Since automation is crucial, we need to discuss what kinds of machine learning algorithms are needed and consider how they may fail, resulting in a need for human involvement. The methodology of photographic censusing is built on analyzing digital images, and, therefore, it is appropriate that computer vision algorithms should be considered primarily for its automation needs. For example, the detection pipeline (discussed in Chapter 3) is used to automatically filter the collected images into a set of relevant animal sightings. As we will see, however, additional computer vision algorithms are also needed to identify potential matches and verify pairs of annotations. Furthermore, computer vision algorithms are imperfect, and curating an ID database requires human effort to fill the accuracy gap. As such, there are three specific and systemic challenges encountered during automated ID curation that we must discuss because they drastically increase the need for

human interaction: 1) the proper selection of annotations from the detection pipeline, 2) the correct matching of the intended animals within annotations, and 3) the managing of ID curation and deciding when a human needs to do additional ID verification. Unfortunately, the amount of human effort required can be substantial when poor-performing computer vision algorithms are used and these sources of error are not appropriately addressed. The following sub-sections describe these problems and explain why they dramatically hinder the automation – and therefore the general usefulness – of photographic censusing.

4.1.1 Which Annotations to Select: Comparable Annotations

The first problem we must consider is the selection of which annotations to use for photographic censusing. To do this, we need to understand how annotations are actually used: annotations are automatically ranked to find potential matches, and algorithms and humans verify the resulting pairs of annotations. With this in mind, what can we conclude when a given pair of two annotations fail to match? Does the pair indeed show two different animals? Not necessarily; there are three distinct possibilities for why two annotations may not match, either 1) the detection pipeline failed to filter at least one of the annotations properly (e.g., poor quality or incompatible viewpoints), 2) the annotations were appropriately filtered and actually showed different animals, or 3) the annotations were filtered correctly but are not *comparable*. Therefore, we can only conclude that two annotations truly show different animals if they are both comparable, showing the same areas of distinguishing information that can be compared and contrasted. Since matching is akin to "marking" an animal in a sight-resight study, it is essential to approach and define comparability as a distinct property of an individual annotation. If both annotations in a pair are comparable, a confident and repeatable decision should be possible given enough time to review the pair, regardless of the skill of a particular reviewer. Furthermore, by not considering comparability, we are potentially allowing ambiguity to enter the ID database.

We would prefer to construct an automated photographic census with annotations guaranteed to be comparable for all potential pairings. By definition, an annotation is comparable if it provides enough visual information to a reviewer such that a "same animal" or "different animals" decision is *always* possible. To provide a counter-example, an *incomparable* match can be seen in Figure 4.1, where the two annotations are exceedingly difficult to compare and would likely be set aside for human review. The distinction of needing "enough" visual information to feel comfortable can be challenging to implement in practice. Finding the right combination of required visual features is

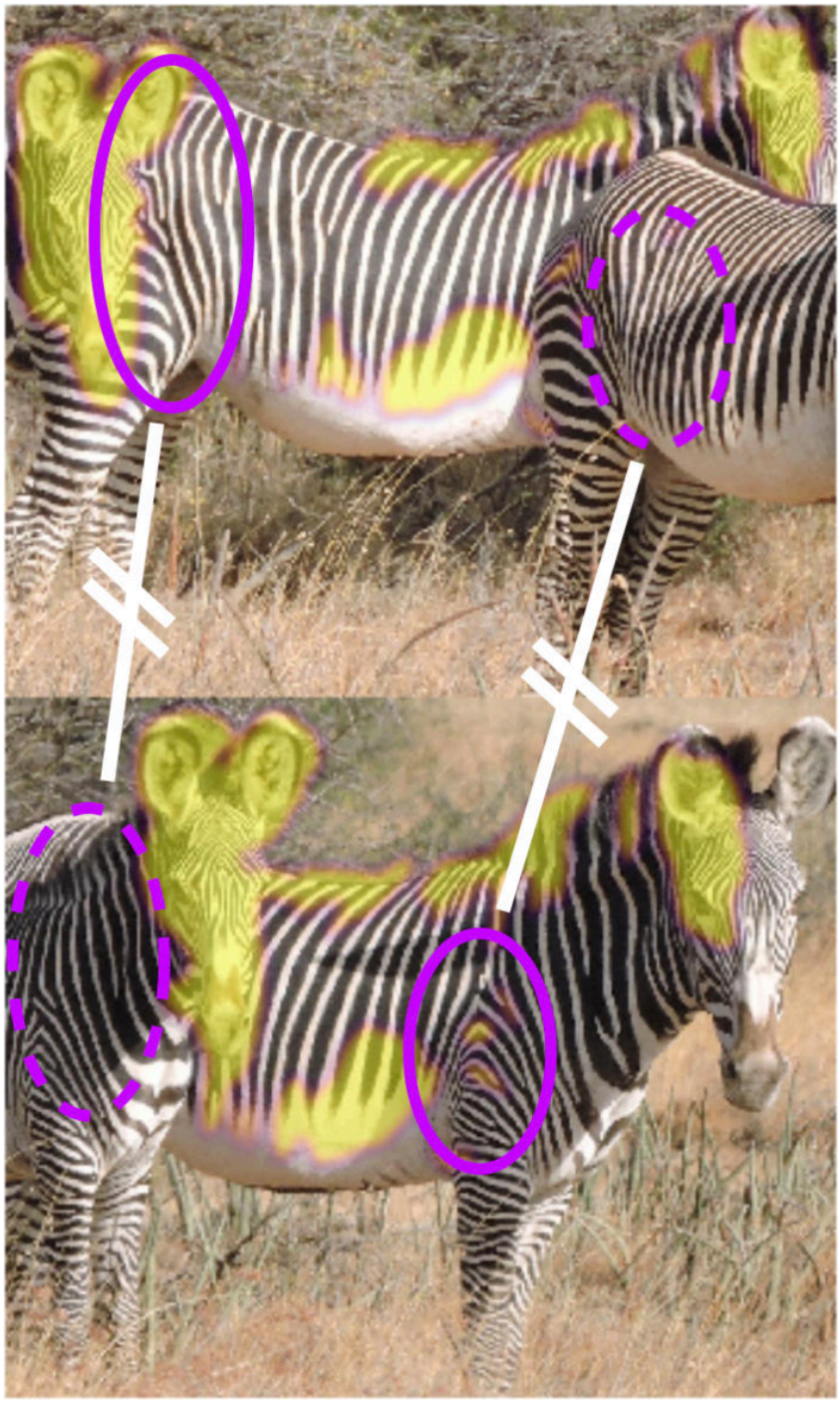


Figure 4.1: An example images of an incomparable match, where the background animals are being compared but a decision cannot be reliably made. The distinctive visual regions that are normally used for verification (the purple oval regions) are both occluded.

subjective and is unfortunately unique for each species. For example, the identifying information for a Grévy's zebra is often concentrated in the back hip region and the shoulder chevron at the top of the front leg (e.g., the purple oval regions in Figure 4.1). Humans commonly use these two areas to verify if a pair of Grévy's zebra annotations are the same or not, and not being able to compare or contrast these regions makes the match much more difficult (or impossible) to decide. This dynamic suggests a trade-off exists between 1) a complete review of all the detected annotations and 2) the desire to increase the automation of the photographic census.

Unfortunately, not all annotations created by the detection pipeline (and properly filtered for relevant species and viewpoints) are guaranteed to be comparable. The need for comparability will be addressed in Chapter 5 where the notion of a Census Annotation (CA)[20] is introduced and added as a new animal detection component. The discussion there will demonstrate that photographic censusing is significantly more automated and maintains the same level of accuracy in its population estimate when only Census Annotations are used.

4.1.2 Systematic Ranking Errors: Incidental Matching

Even with a detection pipeline that can accurately find, label, and identify relevant sightings (and filtering methods like Census Annotation to discern the comparable annotations), it is still possible to have considerable problems during automated ID curation. The second systemic error we must consider happens when a ranking algorithm (see Section 2.3.1) confidently matches two annotations that do not show the same animal. This problem results in errors in the ID database because two distinct animal IDs are then incorrectly and automatically merged. This problem is called "incidental matching'. In order to rely on high amounts of automation during photographic censusing, we need to examine the two most common incidental matching scenarios, photobombs [13] and scenery matches, and review why they happen.

A photobomb happens when the primary animal in one annotation matches a non-primary animal in the other annotation. An example of a photobomb can be seen in Figure 4.2 (left), where the primary animals shown in the middle of each annotation are not being appropriately matched (highlighted regions provided by HotSpotter [261]). Photobombs are typical for herding species because animals that are routinely seen together have an increased likelihood of being seen in the background of other images. Further, a special case of photobombs can also occur between mothers

[20]The motivation of CA is slightly different from Annotation of Interest because comparability does not make an image-level determination and focuses entirely on the annotation. For example, while an AoI may show enough information to identify it *most* of the time, that does not guarantee that it will be universally comparable *all* of the time.

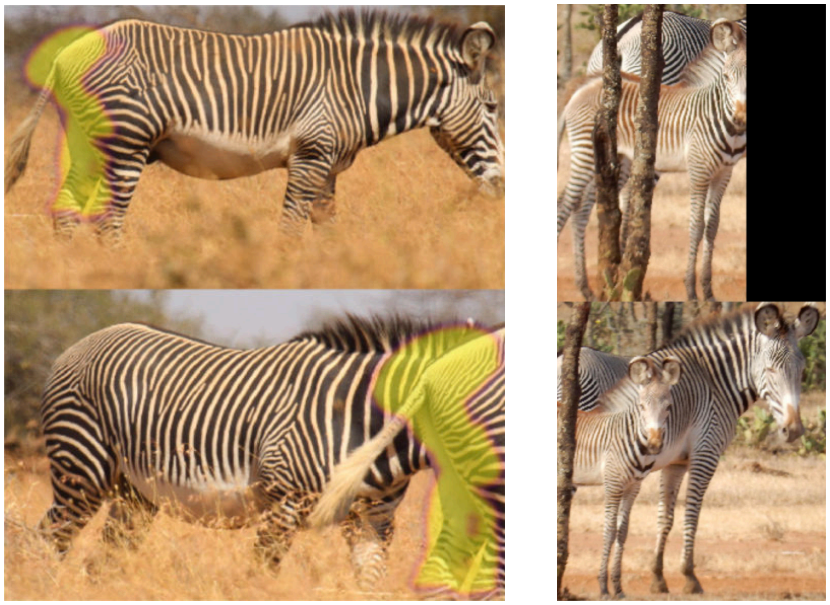


Figure 4.2: Example images of two types of photobombs taken during the GGR-18. A typical photobomb (left) happens when the primary animal in the top sighting has matches against itself in the bottom annotation, but it is not the primary sighting in that annotation. A special case of photobombs, involving splitting mothers and foals (right) in the same image, is particularly challenging to automated ID for herding social species.

and foals, called "mother-foal photobombs". Young animals often stay close to their mothers for protection [375], which means their annotations can significantly overlap and be accidentally matched. Figure 4.2 (right) gives an example of a mother-foal photobomb for Grévy's Zebra during the Great Grévy's Rally (GGR) 2018. The demographics for the GGR 2016 census [356] report that approximately 10% of the zebras in the population were infants (0-12 months of age), meaning that these types of photobombs will happen at a non-trivial rate and must be treated as a distinct error mode.

Scenery matches, in contrast, are distinct from photobombs because they are based on matching the surrounding background (i.e., trees, shrubs). One of the most common sources of incidental matching – especially scenery matches – is annotations taken within seconds of each other; these "near-duplicate" annotations can strongly match because they capture the same scene with significant background overlap. Scenery matches can also occur when multiple photos are taken from the same spot, although not necessarily on the same day [115]. By their very nature as stationary cameras, camera traps have a significant potential for scenery matching and must be considered carefully during a photographic census. Figure 4.3 gives an example of a scenery

Figure 4.3: An example image of a scenery match taken during the GGR-18. The background scene in this match strongly corresponds while the two primary animals are clearly different individuals. Semantic segmentation could provide a background mask but would also require novel ground-truth segmentation data for new species.

match for a reticulated giraffe during the GGR 2018 censusing event. By looking carefully at the giraffe, it is evident that the animals are not the same. However, robust matching is happening by the texture-based algorithm on the background trees and bushes. The result is that this negative "different animal" match will have a strong positive score, complicating our ability to set automatic decision thresholds for pairs in general.

The Census Annotation concept is extended in Chapter 5 to add Census Annotation Regions (CA-R) to avoid incidental matching. A Census Annotation Region is a smaller, more focused box within a CA that crops out irrelevant background information. This new bounding box drastically limits the number of overlapping animals and the amount of background seen within the annotation. The benefit of performing a photographic census on CA-Rs, as we will see, is that it presents highly comparable (and therefore easier) annotation pairs to the automated verifier and vastly reduces the need for human verification to fix bad decisions.

4.1.3 Managing the Decision Process: Animal ID Curation

Now that we have identified the need to filter annotations appropriately and limit the impact of incidental matching, we need to examine the third and final way substantial amounts of human effort are introduced. Let us consider a fully automated process for building an ID database that uses 1) comparable annotations as input, 2) a ranking algorithm to produce potential matches for each annotation, and 3) a verification algorithm to decide if each match is correct. The process assigns the same ID to annotation pairs deemed correct and leaves unmatched annotations as different animals. The immediate question is, "*is this process sufficient to produce an accurate database of animal IDs?*" Consider what would happen when the verifier makes a mistake and incorrectly decides that one of the pairs is the same animal when, in fact, it is not. The IDs for those two animals would be merged, and the population estimate would decrease by one (under-counting). The second type of error in the ID process occurs in one of two ways: if a verifier or human fails to decide "same animal" for an actual match or if the ranking fails to propose the match in the first place. In this event, the ID for one animal is split across two IDs in the database, and the population estimate would increase by one (over-counting).

The natural way to avoid this problem is to rely on human decision-making instead of the automatic algorithm when there is any chance of error. Unfortunately, this solution requires many human decisions – defeating the purpose of automation – and, as we will see, still does not eliminate all chances of error. What is needed is an overarching control algorithm that 1) goes beyond the

matching pairs that were initially suggested by a ranking algorithm, 2) works to find potential inconsistencies in the match decisions, and 3) resolves issues – either fixing inconsistencies or reinforcing recent decisions – by seeking additional information from an algorithm or human. This algorithm must manage when automated or human decisions are needed to curate a consistent database of animal IDs, and its implementation influences how much human decision-making is required overall.

4.1.4 Summary

In summary, these three challenges are the most significant theoretical barriers to automated photographic censusing. The underlying problem is that automated computer vision algorithms and even human reviewers make mistakes. A significant impact of these mistakes – difficult-to-compare annotations, incidental matching, and matching ambiguities – is the increased need for human effort to generate a reliable database of IDs. It is, therefore, appropriate to use human effort as a quantitative metric to compare different censusing configurations. Therefore, the amount of work done by humans will be the basis for experiments in Chapter 5. We now turn our attention to describing the components required to perform a photographic census and discuss the errors they introduce in the final population estimate. The introduced components are assembled at the end of the chapter to build a new evaluation database for Grévy's zebra IDs.

4.2 Components of Animal ID Curation

The task of large-scale and automated photographic animal censusing is complex. The above problem description has established that the solution needs multiple components to work successfully as an end-to-end process. Those components include:

1. a *detection pipeline* that finds relevant sightings of animals in images and ensures that all of the resulting annotations are comparable,

2. a *ranking algorithm* that matches a query annotation against a database of annotations and generates a prioritized ranked list of the most likely pairs that show the same animal,

3. a *decision management algorithm* that looks for ambiguities and inconsistencies in the database and seeks additional pair decisions from a verification algorithm or human reviewer,

4. a *verification algorithm* that automatically predicts if two annotations are the same animal or not,

5. a *human-in-the-loop reviewer* that is tasked by the decision management algorithm to litigate hard pairs when the results of the automated verification algorithm are insufficient, and

6. a *population estimator* that uses the distribution of sightings and resightings on back-to-back days to estimate the total number of animals in a surveyed population.

This dissertation addresses the following aspects of this problem: 1) establishing a problem definition, 2) contributing the detection pipeline, 3) proposing a solution to the annotation comparability problem, 4) minimizing the incidental matching problem, 5) building an experimental dataset, 6) configuring, training, building, and validating algorithms and the whole system, and 7) demonstrating its effectiveness on the GGR-16 and GGR-18 censusing events (Chapter 6). The evaluation employs prior and ongoing work for ranking (HotSpotter [261] and PIE [263]), pair-wise verification (VAMP [13] and PIE [263]), and decision-making during ID curation (Graph ID [13] and LCA). Of particular importance in this work is the first experimental evaluation of the LCA algorithm, demonstrating that it is a successful animal ID curation algorithm. The remaining discussion in this section will review each of the components listed above and outline the errors that each may introduce in the population estimate.

4.2.1 Detection Pipeline

The detection pipeline, as discussed in Chapter 3, functions in part to control which annotations are considered for photographic censusing. By its very design, the output of the detector offers a filtered view to the rest of the photographic censusing procedure because it only allows relevant annotations through to ID. By operating on the reasonable assumption that the input photographs did not capture all of the individuals in a large population, the reality is that photographic censusing must be designed to estimate how many animals were not seen at all. When an animal is not cataloged, it means that 1) the animal was not seen by any photographer or 2) the animal was seen by a photographer but the detection pipeline did not create an annotation for it. Both cases are handled similarly by the population estimator when the chance of a detection error is independent across images and uniformly distributed. For example, we know that the YOLO localizer can perform poorly on small annotations. If the analysis considers only Annotations of Interest, this source of error is nullified because useful and small AoIs are rare. This detail means that the

estimator implicitly controls the error introduced by false-negative detections (missed detections) in the final population estimate.

The more worrying case is when the detection pipeline produces an annotation that it should not have let through. These "spurious detections" may have various issues; they could have a poor bounding box, be the wrong species, show an incomparable viewpoint, be too blurry for matching, be occluded in the background, or are otherwise not identifiable. These annotations are less likely to be successfully matched by an ID ranking algorithm and are therefore more likely added to the animal ID database as singletons. The result is that false positives from detection can bias the number of animals in the population estimate higher. In practice, these errors are easy to find by reviewing all of the annotation singleton (or encounter singleton) IDs in the database. For a little extra human effort to do a final check, this source of error can be mitigated by discarding inappropriate singleton annotations during ID curation.

4.2.2 Ranking Algorithm

After comparable annotations are selected for use in the photographic census, a ranking algorithm is needed to prioritize which pairs of annotations should be reviewed. The ranking process is a crucial step because an exhaustive review of all quadratic pairs is not feasible. An established ranking algorithm discussed and used in this dissertation is the HotSpotter [261] algorithm, which finds texture-based features on the body of an animal and uses a nearest neighbor search database to create a list of ranked matches. The algorithm works well as a general retrieval and ranking method and produces numerical scores for each pair in their ranked lists. These scores, however, are unbounded and do not have a good separation between positive "same animal" pairs and negative "different animal" pairs, setting up the need for an independent verifier (discussed next). There are also new types of ranking algorithms, like Pose Invariant Embeddings (PIE) [263], that use deep learning and specialized training methods (i.e., triplet loss) to learn a feature embedding for ID. These methods are often faster, more flexible, and more accurate compared to their hand-engineered counterparts. The challenge is that these feature embedding approaches can require significant amounts of ground-truth ID data to train, which presents a problem for photographic censusing as an end-to-end process for new animal species. The crucial insight with ranking algorithms is that there needs to be an awareness of the capabilities of traditional computer vision algorithms that do not rely on deep learning and an acknowledgment that they are an asset to bootstrapping.

When using ranking algorithms, the most elusive source of error is a "missed match" (i.e., a

recall failure). The retrieval rates for a ranking algorithm can be estimated using small databases because they are easier to verify thoroughly. It becomes challenging, however, to measure this value for larger databases because a ranking algorithm is often needed to build the database in the first place. Special care is needed, therefore, to ensure that an animal ID database is sufficiently curated with multiple ranking and verification algorithms. The second type of error from a matching failure is a "spurious match", which is most often caused by incidental matching. As we will see, the use of Census Annotation Regions and a decision management algorithm helps to drive this rate towards zero during ID curation. A final consideration is that some animals may be photographed multiple times at the same general time and place – considered a single "encounter" of that animal – and the ranking algorithm needs to support both short-term (intra-encounter) and long-term (inter-encounter) matching. The general expectation is that the rate of accurate matching within an encounter is expected to be higher than between encounters.

4.2.3 Decision Management Algorithm

The management algorithm leveraged in most of this work is called the Graph ID algorithm by Crall [13]. The Graph ID algorithm is a linear curation process that enforces an explicit level of decision redundancy within animal IDs (positive redundancy) and between animal IDs (negative redundancy). The Graph ID algorithm is easy to understand but has the downside that it is too aggressive at enforcing consistency. For example, when an inconsistency is found, the automated verifier is disabled, and manual decision-making by humans is needed to find and fix the issue. Furthermore, the algorithm only uses automated verifiers up to a pre-defined threshold (generally a false-positive rate of 1%). In practice, this means that most of the ID verification decisions are performed by humans. A second decision management algorithm called LCA (Local Clusters and their Alternatives) discards the need for explicit redundancy and instead measures an animal ID's stability relative to an alternative clustering of annotations. LCA intentionally delays human effort as long as possible and uses an automated verifier more effectively by weighting its decisions into a probabilistic vote (compared to a binary decision threshold with Graph ID). The result is that LCA requires much less human effort to resolve inconsistencies and curate the database of animal IDs.

The goal of photographic censusing is to produce a consistent database of animal IDs such that no additional splits or merges need to occur. However, it is significantly more likely to have a silent merge case between the two possible cases because it cannot be identified without some positive signal from a ranking algorithm. On the other hand, missed splits can be identified more easily

by analyzing the state of the database and ensuring each animal ID is sufficiently reviewed. Thus, any bias introduced in the final population estimate is ultimately a matter of how comprehensively the algorithm reinforces the animal IDs, and by default, the estimate should be treated as an upper bound. For example, the decision management algorithm can include a method to "densify" small animal IDs with extra decisions, which helps identify the need for potential ID splits.

4.2.4 Verification Algorithm

The purpose of an automated verifier is to review pairs suggested by the decision management algorithm, and it functions as an accurate stand-in for a human reviewer. The Verification Algorithm for Match Probabilities (VAMP) [13] approach was the first verification algorithm developed for use in automated photographic censusing[21]. VAMP is implemented as a random forest classifier that uses hand-engineered features to compare two annotations and produces a probabilistic decision of "same animal" or "different animals". The algorithm is noticeably fast and, while it does require ID training data, it can be trained from a relatively small database due to a mining procedure. The second type of verification algorithm can be constructed with the same triplet-loss embedding networks used for ranking. Since the embedding for two annotations is trained to be directly compared in embedding space, a distance between two annotations can also be calculated and used for verification. Like with ranking, however, the PIE algorithm needs an extensive database to train effectively; VAMP, in contrast, can work as a bootstrapable verification algorithm before a critical mass of ground-truth IDs can be collected.

The automated verifier's accuracy – and the separability of its scores – directly affects how automated photographic censusing can be. Furthermore, any error this component introduces ends up translating directly into the need for more human effort to find and fix inconsistencies in the animal ID database, and its failures should have a minimal impact on the final estimate.

4.2.5 Human-in-the-Loop Reviewer

Likewise, human reviewers are not perfect and make mistakes. Each mistake made by a human requires the decision management algorithm to request more overall work to find and fix the database issues it causes, just as when the automated verifier makes an incorrect decision. Making the matter worse, the pairs given to humans for review are not consistent in their difficulty. Some

[21]The Graph ID algorithm, using the detection pipeline to find comparable annotations along with HotSpotter as its ranking algorithm and VAMP as its verifier, was used as the original algorithmic tool-chain for the Great Grévy's Rally (GGR), as discussed in Chapter 6

pairs are easy to compare and contrast and, as a result, are faster to review. However, when a pair is complex or shows two annotations that are borderline comparable, then the human reviewer spends significantly more time making a decision. Since our goal is to reduce overall human effort, the most obvious way to achieve this is to automate decision-making with verification algorithms. However, an additional way to reduce effort is to focus on annotation pairs that are easy and fast for humans to review. The error introduced by human reviewers, like the automated verifier, also converts to more overall work during ID curation. This feature of photographic censusing is convenient because it suggests that a comprehensive pairwise review attenuates the effects of human mistakes in the final population estimate.

It is important to note that if humans cannot accurately verify the results of a ranking algorithm, then that algorithm or species is not compatible with photographic censusing. Further, a person must be able to manually filter out annotations that are of undesired species, of incompatible viewpoints, of incomprehensible quality, and are ultimately *incomparable* because all of these are the actions needed to bootstrap the detection pipeline and curate IDs in the first place.

4.2.6 Population Size Estimator

Lastly, we need to highlight that an animal ID database does not provide a population estimate by itself. The animal population is likely open, and the number of known animals in the database does not tell us anything about many *unknown* animals still remain in the population. Put simply, how do we know if an ID database is complete and has 100% coverage? Any animal database for an open population will have the possibility of animals that have never been cataloged. What is needed is a way to use a curated and consistent animal ID database to estimate the total number of animals in the population. The sampling method used by sight-resight studies, the Lincoln-Petersen estimator [376], is a relatively simple ratio calculation. Its simplicity has made it a popular method by ecologists for baseline studies, and it is used for photographic censusing because it allows for more direct comparisons with historical estimates. One advantage of large-scale photographic censusing is that it is designed to be a drop-in replacement for past, more limited surveys.

The Lincoln-Petersen (LP) estimator can be extended for our use with machine learning algorithms. One of the advantages of machine learning algorithms is that their error rates can be experimentally measured during validation, and the effects of automatic failures in the final population estimate can be considered. The various sources of error discussed in this chapter are limited to specific scenarios around missing and making spurious detections, failing to recall

matches during ID ranking, and incidentally matching annotations. An updated version of the LP estimator is proposed in the next section that adds new error terms and discusses their impact on the final population estimate.

4.3 Automated Lincoln-Petersen Estimator

While the Lincoln-Petersen estimator is well established and has been expanded since its formation, it has not been amended to add explicit terms for automated machine learning errors. This section provides a mathematical framework for estimating an animal population using the automated tools and concepts proposed above for photographic censusing. To provide a quick overview: the Lincoln-Petersen estimator and confidence interval (CI) can be modified to add four high-level error rates: missed detections, spurious detections, missed matching, and spurious matching. The estimated rate of missing a match (recall failure) and incorrectly matching two animals together (incidental matching) impact the population estimate the most. In contrast, the rate of missing detections has the most significant impact on confidence interval. One of the convenient takeaways is that aggressive annotation filtering (artificially high detection miss rate) should have little impact on the actual predicted estimate. Further, spurious detections are easy to identify and eliminate during ID curation, ideally a rate of zero in practice by reviewing singletons. Likewise, when the ranking algorithm fails to match at a higher rate than it makes spurious matches, then the population estimate will, as expected, be biased high, and the CI will grow. For readers who wish to skip the details of its derivation, the final resulting Equation 4.21 is used in Chapter 6 to produce a population estimate for Grévy's zebra in Kenya. Section 4.4 continues the discussion by describing an evaluation dataset for Census Annotations in Chapter 5.

4.3.1 Assumptions

The population estimate applies to a fixed time window, where images of animals are taken on day 1 and again for a subsequent, consecutive day 2. The number of animals sighted on each day and the number sighted on *both* days (resights) provides the foundation for a sight-resight study. However, for this process to work accurately, it must rely on a set of assumptions about the data collection and underlying animal detection and identification algorithms.

1. **Equal Sightability** - The animal population is considered closed (geographically and demographically) during the two days of the census; no significant immigrations, emigrations,

births, or deaths should occur, and the actual number of animals is expected not to fluctuate during the survey period [377].

2. **Passive Influence** - Any given animal, or any given group of animals, is equally likely to be sighted throughout any given day, and across both days, of the census; seeing an animal does not affect its likelihood of being seen later that day or resighted on day 2.

3. **Encounter Coverage** - When an animal or a group of animals is encountered, *exactly one* comparable annotation is captured by the photographer(s); this assumption requires that if a photographer encounters an animal, then it cannot be skipped over; another way to phrase this is matching within an encounter of comparable sightings is trivial and assumed to have perfect recall.

4. **Population Coverage** - The total number of animal sightings on day 1, day 2, and the number of resightings on each day and between both days are non-trivial (i.e., not zero or close to zero); the assumption specifies that the scale of data collection is sufficiently large and that the ID database offers meaningful and uniform coverage over the animal population.

5. **Comparable Sightings** - The animal species must be comparable; any given pair of annotations must be verifiable by a human (with high confidence and accuracy) to be either a) same individual or b) different individuals.

6. **Match Retrieval** - The probability of the ranking algorithm failing to retrieve (recall) a correct "same animal" match between encounters is assumed to be non-zero but also constant throughout the census, regardless of the size of the underlying animal ID database.

These assumptions should be realized through a careful design of the data collection procedure, which will be the focus of Chapter 6 and *photographic censusing rallies*. For example, the data collection process should ensure that the census area is comprehensive and covers the known regions that contain the resident population.

4.3.2 Animal Detection

A photographic census is a process where images are captured in a two-day collection and processed by an automated system. The first step of this automated processing is detection, and it places a box and species label around each animal in the collected images. Let $\alpha \in \{1, 2\}$

signify either the 1st or 2nd day of the census, where each has a set of images that can be treated independently. For each day, there are the following:

s_α — total number of animal sightings of the desired species captured by the images

s'_α — total number of *comparable* animals within s_α, where $s'_\alpha \leq s_\alpha$

d_α — total number of animal detections of the desired species derived from the images

Since we only care about the animal sightings that are visually comparable, we can set the expected number of annotations a_α to be:

$$a_\alpha = (1 - p_{dm}(\theta)) * s'_\alpha + p_{ds}(\theta) * d_\alpha$$

where

$p_{dm}(\theta)$ — probability of missing an animal detection, given θ

$p_{ds}(\theta)$ — probability of adding a spurious, incomparable detection, given θ

θ — detection parameter controlling the level of annotation filtering (4.1)

The detection probabilities above aggregate the chances of making a detection error into a single value regardless of the reason. The reasons for making a detection error vary and include factors like qualitative properties of the image (e.g., illumination, sharpness) and semantic properties (e.g., an animal is truncated, occluded, or of a commonly confused, visually similar species) but can be estimated for an entire dataset. The variable θ represents the collection of parameters that control the filtering level applied on the annotations passed to ID. These parameters influence the error rates of the detector because – for example – in focus, clear, un-occluded, and well-lit annotations are easier to find, and the system will make more mistakes as poorer annotations are included. Furthermore, the error rates must use comparable annotations as the frame of reference for correct detections.

4.3.3 Individual Identification on Day 1 and 2

Provided with the number of *comparable* annotations a_α for day α, we need to estimate the number of unique individuals n_α that was sighted on that day. The number of individuals

$n_\alpha \leq a_\alpha$ but, in general, we expect $n_\alpha << a_\alpha$ to be the case for a large-scale photographic census with sufficient coverage. An animal's average number of resightings is expected to be non-trivial and is further guaranteed by assumption 4. Therefore we need a general matching algorithm that can prioritize the potential $\binom{a_\alpha}{2}$ pairs of annotations for analysis and decide if the annotations are either a) the same individual or b) different individuals. All pairwise decisions are guaranteed to be decidable through assumption 5 because an incomparable pair cannot exist. Furthermore, any incomparable pair that is found needs to be manually reviewed to discard the offending annotation(s). This manual review means that each of the decisions has a chance of being decided correctly or incorrectly. An incorrect match could happen by either missing a match (failing to associate two annotations of the same individual correctly) or making a spurious match (failing to distinguish two annotations of a different individual correctly).

In practice, however, it can be too complex to estimate the pairwise error probabilities of missing a given match. The reasons for this are varied and are often a consequence of the implementation details of the matching algorithm. For example, the naïve search space for the match pairs is $O(n^2)$, but a given matching algorithm may choose to perform an approximated search of this space on visually similar neighbors, meaning not all pairs will be explicitly reviewed. Thus, there is a probability that the matching algorithm may fail to include a correct pair in the decision process. However, this probability is complex and may not be the only factor in missing a particular correct match. Therefore, we would prefer to contextualize the matching problem not in terms of a *pairwise* probability of matching failure but rather in terms of a *global* probability of matching failure, averaging the various effects and relying on a general performance validation of the algorithm. Further, we expect that these rates of ID failure are constant and do not depend on the size of the search database (assumption 6). Therefore, we set:

$$n_\alpha = n'_\alpha * (1 + p_{mm}(\theta) - p_{ms}(\theta))$$

where

$$\begin{aligned} n'_\alpha &\text{ — actual number of individuals captured by } a_\alpha \\ p_{mm}(\theta) &\text{ — probability of missing a match, given } \theta \\ p_{ms}(\theta) &\text{ — probability of making a spurious match, given } \theta \end{aligned} \tag{4.2}$$

We would prefer, however, to define n'_α in terms of a_α. Let us assume that there exists some $k_\alpha \in \mathbb{R}^+$

which represents the average number of annotations per individual on day α, where:

$$n'_\alpha = \left\lfloor \frac{1}{k_\alpha} * a_\alpha \right\rfloor \tag{4.3}$$

Assumption 3 guarantees that when an individual photographer (or a group of photographers in the same survey vehicle) encounters a single animal (or a group of animals), then the photographer(s) act in unison as a single oracle. The "photographic oracle" is expected to capture *exactly one* comparable annotation for *every* individual in the encounter, and there will be multiple independent oracles during a photographic census. This assumption is exceedingly strong about a_α and places an unrealistic expectation on the level of skill and coordination between the photographers. Let us consider a relaxation for the moment where more than one annotation for an animal was allowed to be collected but still required for every animal. This second requirement becomes easier to meet when there are multiple photographers in the same survey vehicle. This condition also would mean multiple annotations were taken at the same general time and place for the same animal. Let a'_α represent the total number of comparable annotations that were collected before any sort of de-duplication is performed for each encounter, where $a_\alpha \leq a'_\alpha$.

The difference between a_α and a'_α is that the only way to get a repeat annotation of an individual with a_α is by photographing it at a different encounter. In other words, the value for k_α is equal to the average number of encounters an animal is seen on day α of the census. Each encounter is limited to a fixed spatio-temporal context with a unique duration, interval, and geographic area; the number of individuals seen during each encounter will also vary. Each encounter, however, is a specific event and occurs independently of all other encounters. Assumptions 1 and 2 guarantee that the movement and behavior of the animals are truly independent of the ongoings of the census and that the chance of sightings is uniform. We can also reason that the chance a given group of animals is encountered by two oracles simultaneously is practically zero. However, in any such event, the oracles are simply merged for that single encounter. Since there is no real-time coordination between photographic oracles (which is allowed to encourage better coverage of the survey area), then the process of any oracle encountering any given animal can be modeled as a Poisson random process. This process has an expected value k'_α for the average number of annotations per individual and is conditioned on day α because the averages can differ day-by-day.

Thus, the expected number of annotations per animal is defined as:

$$\begin{aligned} \lfloor k'_\alpha * n'_\alpha \rfloor &= a'_\alpha \\ n'_\alpha &= \left\lfloor \frac{1}{k'_\alpha} * a'_\alpha \right\rfloor \\ \text{where} & \\ k'_\alpha &\in \mathbb{R}^+ \\ \text{and} & \\ k'_\alpha &\geq k_\alpha \end{aligned} \tag{4.4}$$

What is needed is to bridge the gap between Equations (4.3) and (4.4) by substituting k'_α and a'_α with their de-duplicated counterparts. In order to do this, the data collection procedure needs to ensure that 1) the average number of annotations per animal is uniform within an encounter and 2) that the average number of annotations per animal is uniform throughout the day. A real-world example of collection bias is when an encountered group of animals contains an infant. We can reasonably expect that the number of photographs for an infant will be biased arbitrarily high compared to an adult in the same group. The reason is simple: baby animals are cute. This effect can also be seen when an individual photographer encounters a new species and wants to capture many images of it. Another anticipated source of bias is that the chance of taking a picture for a given individual is not consistent throughout the day, in a phenomenon called "photographer fatigue". An initial excitement defines this fatigue at the start of the census rally, resulting in more images taken during the first couple of encounters. As the day progresses, the photographer becomes tired, and the average number of annotations per animal decreases.

Three important factors can help control for these kinds of biases. The first is to have multiple photographers within the same survey vehicle. Having multiple cameras in a car will average out the effect of a single photographer and make their unified photographic oracle more uniform in its behavior. The second is that the photographers should be trained specifically on these two effects (excitement and fatigue) and be encouraged to act consistently throughout the day. The third is that the encounter should contain at least one comparable annotation for each encountered animal regardless of its underlying distribution of duplicate sightings. Assumption 3 specifies that the recall rate of ID ranking is 100% within an encounter, which is reasonable because a much smaller

context is being compared for duplicates (not across the entire census, but just at one time and space). Furthermore, the chance of missing a match or making a spurious match should be the same as the chance of making an error between encounters, meaning the terms can be safely substituted. Combining Equations (4.1), (4.2), and (4.3) gives us the estimated number of individuals on day α as a function of the number of annotations:

$$\begin{aligned} n_\alpha &= n'_\alpha * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \\ &= \left\lfloor \frac{1}{k_\alpha} * a_\alpha \right\rfloor * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \\ &= \left\lfloor \frac{1}{k_\alpha} * [(1 - p_{dm}(\theta)) * s'_\alpha + p_{ds}(\theta) * d_\alpha] * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \right\rfloor \end{aligned} \tag{4.5}$$

Equation (4.5) relies on several error probabilities that can be estimated for the various automated components. Notably, none of them are conditioned on α, implying that the performance of a given algorithm does not depend on exactly when an image was captured. We can safely assume this to be the case because the rally has fixed start and end times and (at least for the GZGC and GGR) occurs during the same daytime hours. The equation does, unfortunately, rely on knowing the actual number for s'_α. Humans could manually review the images to obtain their exact values, but a random sampling of all collected should be sufficient to estimate it. Given a reviewed subset (e.g., 10%) of the complete set of comparable sightings (s'_α annotations) and the total number of detections (d_α), each of the error probabilities can be estimated. Furthermore, the estimated rate of missing a detection can also be re-parameterized on d_α so that the detector error rates are directly comparable and are calculated as a function of the detector's output. In other words, for each correct bounding box that is predicted, we can calculate the rate a second box for a missed comparable annotation should have been predicted (on average). Thus, the estimated miss detection rate can be re-defined such that:

$$(1 - p_{dm}(\theta)) * s'_\alpha \approx (1 - \hat{p}_{dm}(\theta)) * d_\alpha \tag{4.6}$$

Reformulating Equation (4.5) with the respective substitutions for the estimated error probabilities, and using Equation (4.6), results in:

$$\begin{aligned} n_\alpha &= \left\lfloor \frac{1}{k_\alpha} * [(1 - p_{dm}(\theta)) * s'_\alpha + p_{ds}(\theta) * d_\alpha] * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \right\rfloor \\ &\approx \left\lfloor \frac{d_\alpha}{\hat{k}_\alpha} * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta)) \right\rfloor \\ &\approx \lfloor \hat{n}_\alpha * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta)) \rfloor \end{aligned} \tag{4.7}$$

where

$\hat{n}_\alpha$ — number of animals in the ID database seen on day α

4.3.4 Individual Identification Between Days 1 and 2

Given the number of individuals seen on day 1 and day 2, represented by n_1 and n_2 from Equation (4.7), we now need to model the number of individuals n_B that were seen on *both* days. Following the same logic as Equation (4.5), we let:

$$\begin{aligned} n_B &= n'_B * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \\ &= \left\lfloor \frac{1}{k_B} * a_B \right\rfloor * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \end{aligned} \tag{4.8}$$

We need to recall Equation (4.1), however, to consider the total number of annotations that are involved. Before, when considering only one day at a time, we could estimate the probability of missing the detection for a given day as a single event. The value for a_B, however, is based on a joint probability of a successful detection (and successful comparable decision) in both day 1 *and* in day 2. Luckily, these joint probabilities are independent and can be combined. Furthermore, it is incredibly unlikely to match spurious detections across days (regardless of θ) and it can be assumed that $p_{ds}(\theta) = 0$ for resightings. The value for a_B is therefore defined as:

$$\begin{aligned} a_B &= (s'_1 + s'_2) * Prob(\text{detected on day 1} \cap \text{detected on day 2}) \\ &= (s'_1 + s'_2) * Prob(\text{detected on day 1}) * Prob(\text{detected on day 2}) \\ &= (s'_1 + s'_2) * (1 - p_{dm}(\theta))^2 \end{aligned} \tag{4.9}$$

Substituting Equation (4.9) into Equation (4.8), and applying the same substitutions used in (4.5), results in:

$$
\begin{aligned}
n_B &= \left\lfloor \frac{1}{k_B} * a_B \right\rfloor * (1 + p_{mm}(\theta) - p_{ms}(\theta)) \\
&\approx \left\lfloor \frac{d_1 + d_2}{\hat{k}_B} * (1 - \hat{p}_{dm}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta)) \right\rfloor \\
&\approx \left\lfloor \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta)) \right\rfloor
\end{aligned}
\tag{4.10}
$$

where

$\hat{n}_B$ — number of animals in the ID database seen on both days

4.3.5 Animal Population Estimation

We now have the estimated total number of individuals seen on day 1 (n_1), day 2 (n_1), and the number of individuals sighted on both days (n_B). Let $n_T \in \mathbb{Z}^+$ be the actual total number of individuals in the animal population. We would also like to estimate n_T as well, given the ratio of animals that were resighted (n_B) by calculating its most likely value. The likelihood function for n_T is sampled from a hypergeometric distribution and, since we are sampling (without replacement) from a fixed pool of individuals, is defined as:

$$
L(n_T|n_B) = \frac{\binom{n_1}{n_B} * \binom{n_T - n_1}{n_2 - n_B}}{\binom{n_T}{n_2}} \tag{4.11}
$$

To find the most likely population estimate $\underset{n_T}{\operatorname{argmax}}\ L(n_T|N_B)$ we must measure the increase in likelihood from n_{T-1} to n_T since we cannot integrate it directly as a non-continuous integer value. We must instead maximize the ratio of likelihoods as:

$$
\begin{aligned}
\frac{L(n_T|n_B)}{L(n_T - 1|n_B)} &= \frac{\frac{\binom{n_1}{n_B} * \binom{n_T - n_1}{n_2 - n_B}}{\binom{n_T}{n_2}}}{\frac{\binom{n_1}{n_B} * \binom{n_T - n_1 - 1}{n_2 - n_B}}{\binom{n_T - 1}{n_2}}} \\
&= \frac{(n_T - n_1) * (n_T - n_2)}{n_T * (n_T - n_1 - n_2 + n_B)}
\end{aligned}
\tag{4.12}
$$

The above ratio exceeds 1 if and only if

$$\begin{aligned} n_T * (n_T - n_1 - n_2 + n_B) &< (n_T - n_1) * (n_T - n_2) \\ n_T &< \frac{n_1 n_2}{n_B} \end{aligned} \tag{4.13}$$

Therefore, the maximum likelihood for the population estimate is:

$$n_T = \left\lfloor \frac{n_1 * n_2}{n_B} \right\rfloor \tag{4.14}$$

Equation (4.14) forms the statistical basis of the Lincoln-Petersen [376] estimator when it is assumed that n_T is sampled from a uniform prior distribution (Assumption 1). Furthermore, a confidence interval (CI) can be added to the estimator using the appropriate Weld method, as derived by [12] and [332]:

$$\begin{aligned} n_{LP} &= \mu_{LP} \pm z_{\alpha/2} * \sigma_{LP} \\ \text{where} \\ \mu_{LP} &= n_T = \left\lfloor \frac{n_1 * n_2}{n_B} \right\rfloor \\ \text{and} \\ \sigma_{LP} &= \sqrt{\frac{n_1 * n_2 * (n_1 - n_B) * (n_2 - n_B)}{n_B^3}} \end{aligned} \tag{4.15}$$

The final Lincoln-Petersen population estimate n_{LP} is a value with a likelihood range[22]. For a confidence interval of 95%, we set $z_{\alpha/2} = 1.96$.

[22]The α term here is overloaded in our notation and does not refer to the day α, but rather a proportion of likelihood for the estimate falling outside of the CI.

4.3.6 Population Estimate Mean

Combining Equations (4.7) and (4.10) with (4.15) for the population estimate mean μ_{LP}:

$$\begin{aligned}
\mu_{LP} &= \left\lfloor \frac{n_1 * n_2}{n_B} \right\rfloor \\
&\approx \left\lfloor \frac{\hat{n}_1 * \hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{\hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \right\rfloor \\
&\approx \left\lfloor \frac{\hat{n}_1 * \hat{n}_2}{\hat{n}_B} \right\rfloor * \frac{(1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{(1 - \hat{p}_{dm}(\theta))^2} \\
&\approx \hat{\mu}_{LP} * \frac{(1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{(1 - \hat{p}_{dm}(\theta))^2}
\end{aligned} \tag{4.16}$$

This implies that the final Lincoln-Petersen estimate μ_{LP} is approximated by the estimate $\hat{\mu}_{LP}$ as calculated directly from the number of animals and resightings in the animal ID database. In practice, the parameters for θ can be selected such that the chance of adding a spurious, *comparable* detection to the census is low, with the ID curation process also ensuring that the chance is functionally zero by requiring a final review of all singletons for irrelevant or incomparable annotations. As a result, if the animal IDs in the database are comprised of only relevant and comparable annotations, then $\hat{p}_{ds}(\theta) = 0$. Thus, the ratio between the final and calculated estimates is:

$$\frac{\mu_{LP}}{\hat{\mu}_{LP}} \approx 1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta) \tag{4.17}$$

and indicates that matching errors mainly impact the final population estimate. This derivation is encouraging for our desire to do annotation filtering because it implies that focusing on comparability and automated curation has less impact on the accuracy of the population estimate. When the rate of spurious detections is zero, the bias from unmatched singletons to push the population estimate higher is eliminated. This fact reinforces the need to perform manual verification of the singletons throughout the curation process as the work needed by humans can be directly justified as improving the accuracy of the estimate. In practice, the hope would be that $\hat{p}_{ms}(\theta)$ is small enough to have a trivial effect on the estimate. For example, a goal of the decision management and verification algorithms is to drive the chance of a lingering split in the database to zero with accurate human decisions and internal consistency checks for all animal IDs. This process would leave lingering merges and $\hat{p}_{mm}(\theta)$ as the only significant source of error from machine learning algorithms that

could incorrectly inflate the population estimate.

4.3.7 Population Estimate Confidence Interval

We now wish to combine Equations (4.7) and (4.10) with (4.15) for the population estimate confidence interval σ_{LP}. We will drop the floor notations here for clarity because a) their effect is less significant inside the radical and b) we are already manipulating approximated values.

$$\begin{aligned}\sigma_{LP} &= \sqrt{\frac{n_1 * n_2 * (n_1 - n_B) * (n_2 - n_B)}{n_B^3}} \\ &= \sqrt{\frac{A * B}{C}}\end{aligned}$$

where

$$\begin{aligned}A &= n_1 * n_2 \\ &\approx \hat{n}_1 * \hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))^2 \\ B &= (n_1 - n_B) * (n_2 - n_B) \\ &\approx \Big[\Big(\hat{n}_1 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2\Big)\Big] \\ &\qquad * \Big[\Big(\hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2\Big)\Big] \\ &\qquad * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))^2 \\ C &= n_B^3 \\ &\approx \hat{n}_B^3 * (1 - \hat{p}_{dm}(\theta))^6 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))^3\end{aligned} \tag{4.18}$$

This means that the approximated CI value from Equation (4.18) simplifies to:

$$
\begin{aligned}
\sigma_{LP} &= \sqrt{\frac{A * B}{C}} \\
&\approx \sqrt{\frac{\hat{n}_1 * \hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{\hat{n}_B^3 * (1 - \hat{p}_{dm}(\theta))^6}} \\
&\quad * \sqrt{\hat{n}_1 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \\
&\quad * \sqrt{\hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \\
&\approx \sqrt{\frac{\hat{n}_1 * \hat{n}_2 * (\hat{n}_1 - \hat{n}_B) * (\hat{n}_2 - \hat{n}_B)}{\hat{n}_B^3}} \\
&\quad * \sqrt{\frac{(1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{(1 - \hat{p}_{dm}(\theta))^6 * (\hat{n}_1 - \hat{n}_B) * (\hat{n}_2 - \hat{n}_B)}} \\
&\quad * \sqrt{\hat{n}_1 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \\
&\quad * \sqrt{\hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \\
&\approx \hat{\sigma}_{LP} * \sqrt{\frac{(1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta))^2 * (1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta))}{(1 - \hat{p}_{dm}(\theta))^6 * (\hat{n}_1 - \hat{n}_B) * (\hat{n}_2 - \hat{n}_B)}} \\
&\quad * \sqrt{\hat{n}_1 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2} \\
&\quad * \sqrt{\hat{n}_2 * (1 - \hat{p}_{dm}(\theta) + \hat{p}_{ds}(\theta)) - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))^2}
\end{aligned}
\tag{4.19}
$$

If we apply the same assumptions about θ and the curation process where $\hat{p}_{ds}(\theta) = 0$, then Equation (4.19) above simplifies considerably and the ratio between the final and estimated CI is:

$$\frac{\sigma_{LP}}{\hat{\sigma}_{LP}} \approx \frac{\sqrt{1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta)}}{1 - \hat{p}_{dm}(\theta)} * \sqrt{\frac{(\hat{n}_1 - \hat{n}_B * (1 - \hat{p}_{dm}(\theta))) * (\hat{n}_2 - \hat{n}_B * (1 - \hat{p}_{dm}(\theta)))}{(\hat{n}_1 - \hat{n}_B) * (\hat{n}_2 - \hat{n}_B)}} \tag{4.20}$$

Equation (4.20) implies that the probability of missing a detection most significantly impacts the Confidence Interval (CI) for the Lincoln-Petersen estimator. In summary, the Lincoln-Petersen estimator (with CI) from Equation (4.15) can be extended with Equations (4.7) and (4.10) to support the error rates of machine learning algorithms as:

$$\begin{aligned} n_{LP} &= \mu_{LP} \pm z_{\alpha/2} * \sigma_{LP} \\ &\approx \left\lfloor \frac{\hat{n}_1 * \hat{n}_2 * \beta}{\hat{n}_B} \right\rfloor \\ &\pm 1.96 * \sqrt{\frac{\hat{n}_1 * \hat{n}_2 * \beta * (\hat{n}_1 - \hat{n}_B * \gamma) * (\hat{n}_2 - \hat{n}_B * \gamma)}{\hat{n}_B^3 * \gamma^2}} \end{aligned} \tag{4.21}$$

where

$$\begin{aligned} \beta &= 1 + \hat{p}_{mm}(\theta) - \hat{p}_{ms}(\theta) \\ \gamma &= 1 - \hat{p}_{dm}(\theta) \end{aligned}$$

The equation above extends the standard Lincoln-Petersen estimator by adding three specific terms for machine learning errors. The lack of $\hat{p}_{ds}(\theta)$ in these equations is convenient but comes at the cost of required human work to double-check the validity of infrequently-seen animal IDs. The required error rate estimates and this derivation will be used in Chapter 6 to estimate the final population of Grévy's zebra in Kenya.

4.4 The Grévy's Zebra Census Dataset (GZCD)

Most animal computer vision datasets are primarily concerned with evaluating "classic" machine learning components, such as training and evaluation data for a detector that only relies on annotating boxes for a series of images. This kind of annotation work is easily distributed and is

Table 4.1: The number of images captured in Meru county on two days of the GGR-16 and two days of the GGR-18.

Rally	Day	Date	Images
GGR 2016	Day 1	January 30th, 2016	1,209
GGR 2016	Day 2	January 31st, 2016	1,695
GGR 2018	Day 1	January 28th, 2018	1,331
GGR 2018	Day 2	January 29th, 2018	1,229
TOTAL			**5,464**

quick to verify as the images can be considered in isolation from each other. This linear complexity for detection-style curation is in sharp contrast with the quadratic complexity of ID curation, where a fully manual review of all pairs of annotations in a database quickly outpaces the ability of humans to curate it exhaustively. However, a reliable, ground-truth target for the actual number of animals in a population is needed to properly evaluate the effectiveness of annotation filtering methods and measure their relative impact on the final population estimate. Furthermore, this ID ground-truth needs to be constructed using easy-to-ID and difficult-to-ID annotations of animals because distracting, time-consuming data is what the filtering is designed to minimize. Animal ID often relies on impeccably clean (i.e., exemplar) animal sightings and often entirely precludes poor images from being given ground-truth IDs. The above presents a challenging dataset problem, as there did not previously exist a readily available, reliable, curated, and large-scale photographic dataset for animal ID that offered both relevant *and* compromised data. While there are a few existing animal ID datasets available publicly, they do not offer the high level of robustness and completeness needed to validate the process of photographic censusing fully. The Grévy's Zebra Census Dataset (GZCD) is proposed here to fill this critical gap and will be made available to the research community in the future.

4.4.1 Images & Annotations

The images in the GZCD dataset are sourced from Meru County, Kenya and are taken over four days of the Great Grévy's Rally (GGR) in 2016 and 2018. These two photographic censusing rallies will be described in more detail in Chapter 6 but – to provide a quick summary – each event was designed to be a "snapshot" photographic census of the resident Grévy's zebra population in Kenya. Photographs were taken on two consecutive days in January during each rally, as shown in

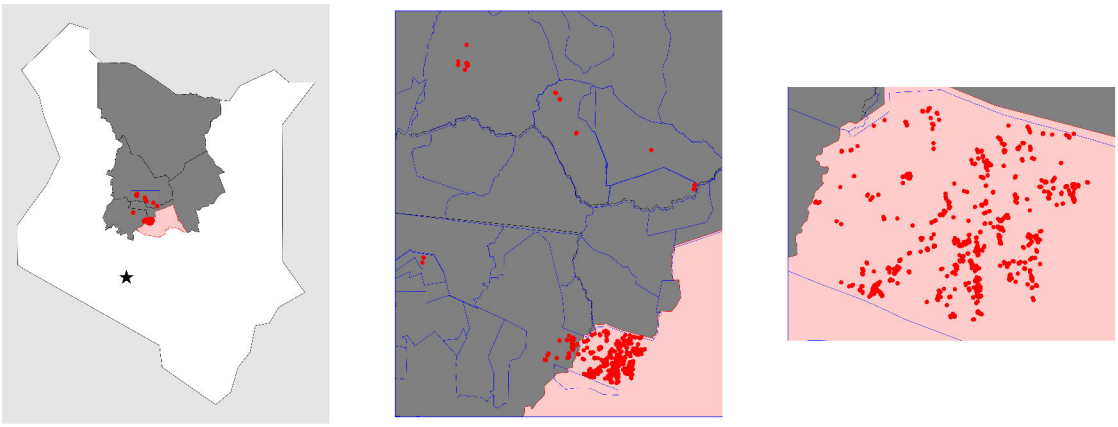


Figure 4.4: The map of image GPS locations in the GZCD dataset. Meru County, Kenya (in red) is located north of the capitol (star) and is at the base of Mt. Kenya. The dataset is comprised of 5,464 images, taken mostly over 4 days (2 days in 2016 and 2 days in 2018), by 13 photographers. Includes all images by photographers that took images in Meru County, even if they were not taken in that county.

Table 4.1. The photographers were trained to capture a consistent viewpoint (right side) for Grévy's zebra and focus on comparable sightings for their eventual ID during both events. The spatial subset for Meru County, Kenya is geographically isolated by mountains from neighboring conservation areas, giving the expectation that the population is largely self-contained [5]. The images in the dataset were taken by 13 photographers (8 from GGR 2016, 5 from GGR 2018). Furthermore, all of the images taken by these photographers were included in the GZGC dataset; images were even included if taken outside Meru County and captured outside the four censusing event days. See Figure 4.4 for GPS locations of the images. In total, 5,464 images were selected for use with the dataset.

The dataset is highly curated; bounding boxes (annotations) and labels (species, viewpoint, and quality) were manually set for all animals to ensure accuracy and consistency. The annotations in the dataset cover 23 unique object classes, ranging from "gazelle" to "car" to "bird" to two different species of zebra. In total, 13,823 annotations were created, of which 9,205 were of Grévy's zebra. On average, 2.5 annotations were created per image, with one image contributing 44 annotations. The photographers correctly followed the instructions to emphasize taking images of the intended sides of the animals (back-right, right, and front-right); human reviewers found a total of 7,372 annotations showing some degree of the right side. Of these, blurry and otherwise poor images were filtered out using a human-labeled quality decision, keeping 4,119 candidate "quality

Figure 4.5: Example annotations from GGR-18 that are poor candidates for identification. These images are fairly typical (i.e. not extreme outliers) and demonstrate the various types of problems encountered by data collection (from top left to bottom right): (top row) occlusion, high amounts of overlap, quality (focus), and (bottom row) truncation, viewpoint/pose, and context.

baseline" annotations for ID review. Figure 4.5 shows example annotations which are not useful for visual ID. Furthermore, relatively poor-quality annotations were also needed to demonstrate that the filtering supplied by Census Annotations (described in Chapter 5) worked to improve matching. The CA classifier for Grévy's zebra was run with an exceptionally low threshold of 0.001 (compared to the recommended value of 0.31) to search for "bad" annotations but still filtered out abject "junk" annotations that showed nothing worthwhile. An additional set of 1,162 low-quality annotations were added for the purpose of evaluating CA, and a resulting collection of 5,281 annotations was sent ID for curation.

The state-of-the-art visual ranking algorithm HotSpotter [261] was used to find different annotations of the same animal. The annotations were visually matched by creating a searchable database of SIFT features and generated results in a ranked list of potential matches. The graph of annotations was fully curated using this ranked list of matches and the Graph ID algorithm [13]. The algorithm uses a series of phases that switches back and forth between adding positive decisions from ranking and ensuring that those decisions are consistent with the state of the graph as an automated classifier and human reviewers make decisions. The algorithm has five distinct phases

and begins with an initial recovery condition (phase 0) to ensure that the current graph has no inconsistencies. An inconsistency is created when a chain of nodes connected by positive decisions also has negative decisions between two of its members (and vise-versa). Once all inconsistencies are manually resolved, phase 1 continues by reviewing a list of match decisions recommended by a ranking algorithm. Each pair was given to VAMP [13] using a pre-trained verification algorithm configured for Grévy's zebra and was either automatically decided using a set threshold or was subsequently given to a human to make a manual decision. This process continues until all of the decisions have been considered, and all inconsistencies are resolved. After the initial ranking, pockets of nodes within the graph form positive connected components (PCC). Phase 2 reinforces all of the PCCs in the graph by ensuring that each node has a set (two, by default) number of redundant positive decisions with another member of the connected component, searching for possible splits. A negative redundancy check is performed in Phase 3 between all of the matched PCCs to ensure sufficient checks for unresolved merges. Finally, the algorithm converges in Phase 4 when all (or a statistically significant number) of the PCCs satisfy the positive and negative redundancy requirements, and its ID graph is free of inconsistencies.

In summary, 5,281 zebra annotations were clustered into 554 names, using 24,129 automated and 43,048 human pair decisions. This work required thousands of hours of review by an external team of human reviewers. Each annotation participated in 25.7 decisions on average (σ=34.3). Thus, the image data collected in Meru County during the two GGR events represented an ideal dataset for evaluation since it is self-contained, contains a sufficiently large population, and is highly curated.

4.4.2 ID Curation & Accuracy Verification

Over 67,000 pairwise decisions were made using the Graph ID algorithm [13] to ensure that each cluster of annotations was internally consistent, representing a comprehensive (but not entirely exhaustive) review of the population graph. The Graph ID required *each* annotation within an animal ID (representing a single named zebra) to have at least two positive decisions with other annotations in the same cluster, whenever possible. This redundancy requirement ensured that a given name did not incorrectly contain multiple animals by mistake, otherwise requiring a split and increasing the number of names. We also configured the algorithm to require each animal ID to have at least two negative decisions between all other matched IDs. This requirement ensured that a given cluster did not need to be combined with another. Otherwise, the database would require a merge, and the number of names would decrease. In total, each annotation, on average, participated in 5.5

positive and 20.0 negative decisions. The effort required to create this dataset involved hundreds of hours of human labeling, months' worth of review, and over $10,000 in direct payments for external labor.

The ML-based automated decision classifier used for approximately 40% of the decisions was not a perfect oracle. The algorithm (VAMP [13]) was configured to allow positive and negative decisions if the predicted confidence was above a threshold, 0.7732 and 0.8605, respectively. Based on cross-validation results, these thresholds were selected with validation data to allow an FPR $\leq 1\%$. Since the Graph ID algorithm enforced redundancy and ensured consistency, there are likely to be few name errors resulting from using this automated classifier. The chance of systematic errors is even less likely because each annotation participated in 16.3 (σ=28.6) human decisions on average.

The experimental LCA algorithm was then run on the final clusters produced by Graph ID to identify possible ground-truth errors and further validate the GZCD. The PIE triplet-loss algorithm was also trained on the ground-truth IDs on Census Annotation Regions. The verification of the ID database was performed in multiple stages of review, using the following combinations: Graph ID with HotSpotter as the ranker and VAMP as the verifier (legacy configuration), LCA with HotSpotter and VAMP, LCA with HotSpotter and PIE, LCA with PIE and VAMP, and LCA with PIE as the ranker and PIE (as well) as the verifier. Each round of verification simulated the final state of the database using the ground-truth human decisions already in the database. When the converged animal ID database was calculated, it was compared to the existing ground-truth database, and differences were examined by hand. The first comparison identified 37 ID updates that performed 14 splits, 13 merges, and corrected 10 miscellaneous labeling errors. The split cases were identified as challenging photobomb examples, which had animals seen together on multiple occasions and 1 mother/foal example (originally VAMP false positives). A benefit of using this additional check is that some animal IDs were suggested to be split by removing only one annotation, which identified inappropriate (i.e., poor quality) annotations that needed to be discarded. The merge issues found by the PIE ranking algorithm mainly were concerned with adding a singleton ID into a larger ID for the same animal (Graph ID retrieval errors). It was apparent during a review that some errors were caused by poor quality annotations or had a very oblique angle (ground-truth labeling error). Once all animal IDs were fixed in the database, no further corrections were identified from subsequent simulations and a manual review of their suggested changes. In summary, the final GZCD database is found to be consistent by the following: two detection filtering configurations

(baseline quality annotations and CAs), two separate ranking algorithms (a hand-engineered feature and learned embedding), two decision management algorithms (one that ensures positive/negative redundancy and another on cluster stability), two verification algorithms (random-forest classifier and a distance-based embedding), and over 43,000 human verification decisions of annotation pairs.

Any ID analysis of the number of annotations and names must also consider *encounters*. An *encounter* is defined as an animal that is seen during a single *occurrence*. An occurrence is the collection of all images that are taken at the same location and time. Another way of phrasing this: if a photographer took ten pictures of the same animal within one second, that would result in ten unique annotations but still only one encounter of that animal. If ten photographers each took ten pictures of one animal during the same occurrence, that would result in 100 annotations but still only one encounter of that animal. This analysis is intended to identify how many unique encounters an animal has because it represents a more accurate semantic understanding of how many times an animal is truly *resighted*. For a population estimate, the number of animals that were *sighted* (or encountered) on days 1 and 2 of a given census can be calculated and compared to the number of animals that were encountered on *both* days (resightings).

The number of occurrences was calculated using an agglomerate clustering of GPS coordinates and EXIF times provided by the GPS-enabled cameras. We assumed a maximum speed of 2 m/s for a walking zebra and required that images be taken within 10 minutes. The dataset offers 296 unique occurrences and 1,803 encounters of zebras. The average occurrence has 25.1 named zebra annotations but only 8.6 encounters, indicating that each encounter contains around 2.9 annotations on average. The amount of review redundancy for encounters is dramatic, with 51.2 total decisions on average. In summary, the dataset contains 554 uniquely named zebras, with 9.5 (σ=8.4) annotations and 3.3 (σ=2.2) encounters per name on average. There are 95 names that contain only one annotation (and trivially only one encounter) and are referred to as *singletons* against the remaining 459 *multitons*. The most often-seen animal had 49 annotations and 14 encounters. When viewing names within encounters, there are 172 "encounter singletons" (up from 95 annotation singletons) and 382 "encounter multitons" (down from 459 annotation multitons).

For GGR 2016, 263 and 334 unique names were seen on days 1 and 2, respectively, and 219 on both days. For GGR 2018, 312 and 302 unique names were seen on days 1 and 2, respectively, and 210 on both days. Using the standard Lincoln-Petersen estimator, the zebra population is calculated to be 402.0 ± 32.0 in 2016 and 449.0 ± 34.0 in 2018 with a 95% confidence interval. Thus, the dataset captures 94.0% of Grévy's zebra in 2016 and 90.0% in 2018 for Meru county

when comparing the total number of names seen each year. Further, 228 animals were resighted across the two-year census gap, and 107 were sighted on all four days. These numbers indicate that the resident population of zebra in Meru County is highly stable over time – over 50% of the expected population in 2016 was resighted in 2018. Thus, the expected population estimates indicate a healthy 12% population growth rate over two years for Grévy's zebra in Meru County, Kenya.

4.5 Summary

This chapter introduces the concept of photographic censusing for the problem of large-scale animal population censusing. The methodology is designed as an end-to-end process. It uses machine learning components for automation, including: 1) a detection pipeline to find relevant and comparable annotations, 2) a ranking algorithm to search for visual matches, 3) a decision management algorithm to control how and why human work is needed, and 4) a verification algorithm to automate the review of pairs of annotations. Beyond these components, the procedure uses human-in-the-loop reviewers and a population estimator to produce a final population estimate. The population estimate generated by photographic censusing is calculated by the Lincoln-Petersen estimator, which is extended here to add new terms for machine learning errors. Lastly, a significant contribution of this chapter is the creation of a large ID dataset (GZCD) for evaluating animal detection, animal identification, and – most importantly – the end-to-end censusing process. The next chapter will use this evaluation dataset to show how finding comparable annotations is crucial to automation.

CHAPTER 5
CENSUS ANNOTATION

Automated photographic censusing is sensitive to how its annotations are selected and why they have matched. When the ID curation process encounters incomparable pairs or incidental matching, it can create errors in the ID database that must be fixed with human interaction, as we have discussed. Fortunately, the adverse effects on automation and accuracy can be mitigated when 1) the ID curation process only considers comparable annotations and 2) the automated ranking algorithm is forced to compare the appropriate visual information between annotations. Let us consider a real-world example shown in Figure 5.1 on the next page, with two images (top and bottom) of Grévy's zebra (*Equus grevyi*). Of the 13 animals seen across both images, only four are valuable to photographic censusing (highlighted as red and blue annotations) because they are universally comparable. The remaining nine annotations (in dashed green) are either incompatible or too challenging for censusing as they show incorrect viewpoints[23], are truncated by the edge of the image, or are significantly occluded by other animals or brush. The middle row of Figure 5.1 shows two comparable annotations (red and blue outline) of the same animal, with matched patterns highlighted in yellow (within the purple dashed ovals) from the HotSpotter [13] algorithm. If all purple oval regions were artificially blacked out, verifying the pair by hand would be substantially more difficult and would take a human more time to make an accurate decision. Furthermore, we recognize that the visual information outside the purple rectangles for each annotation is not critical to this verification task. The pixels outside the purple boxes could have been safely omitted to focus the verifier's attention on the matched area without impacting the ability to make a "same animal" decision.

New detection methods are needed that can 1) quickly determine if an annotation is comparable and 2) remove distracting background information by locating the area that is most likely to match for that species. This chapter introduces the concepts of "Census Annotation" (CA) and "Census Annotation Region" (CA-R) to address these two problems, and it proposes adding two new machine learning components to the existing detection pipeline:

1. **Census Annotation (CA)** - a binary classifier that determines if an annotation contains a comparable region and produces a confidence score, and

[23]The decision on a preferred viewpoint is arbitrary but needs to be consistent throughout a census. The right side was preferred for all image collections with Grévy's zebra and reticulated giraffe (*Giraffa reticulata*). These two species have different patterns on their left and right sides and thus cannot be compared across differing viewpoints.

2. **Census Annotation Region (CA-R)** - a regression network that restricts an annotation's existing bounding box to only include the comparable region for a given species.

An annotation is determined to be a Census Annotation if and only if a valid Census Annotation Region can be drawn inside its existing bounding box. Furthermore, A Census Annotation Region[24] must contain enough visual ID information to prevent the possibility of an incomparable match decision (for a given ranking algorithm). Thus, by definition, if a CA-R bounding box cannot be drawn for an annotation, it cannot be a CA. Furthermore, if two CA Regions are incomparable, at least one of the annotations is not a valid CA. For example, the hip and shoulder chevron (purple dashed ovals in Figure 5.1) are required for Grévy's zebra annotations to be considered comparable. Missing either of those two areas makes the annotation less likely to reliably match, increasing the possibility of an incomparable decision and the ultimate need for human intervention. To prevent this, the CA-R region for that species must clearly contain both of these regions.

The remaining sections of this chapter present a description of the methods and an analysis of both Census Annotations and Census Annotation Regions. First, a dataset for training both of these components is presented for Grévy's zebra and reticulated giraffe, which were the species of interest for the Great Grévy's Rally in 2018 (discussed in Chapter 6). A user study is also performed on how accurately and quickly humans can distinguish match pairs between "normal" detected annotations, CAs, and CA-Rs. Using CAs and CA-Rs with ID curation also positively impacts the performance of automated verifiers, and analysis on training stability and score separability is provided. The discussion then turns to incidental matching and how CA-R can reduce the rate of photobombs and scenery matches by eliminating the background information that causes these types of errors. Lastly, automated simulations are used to measure how much human effort is needed to curate an animal ID database from scratch, with and without CA and CA-R, and demonstrate two crucial results: 1) using CAs and CA-Rs during ID curation results in a substantial increase in automation, 2) the population estimates produced when only CAs and CA-Rs are considered are consistent with the estimate generated when a more comprehensive set of annotations from GZCD (described in Section 4.4) are used.

[24]Census Annotation Regions may be referred to as "CA-R" or "CA Regions" in this discussion, as needed, for clarity.

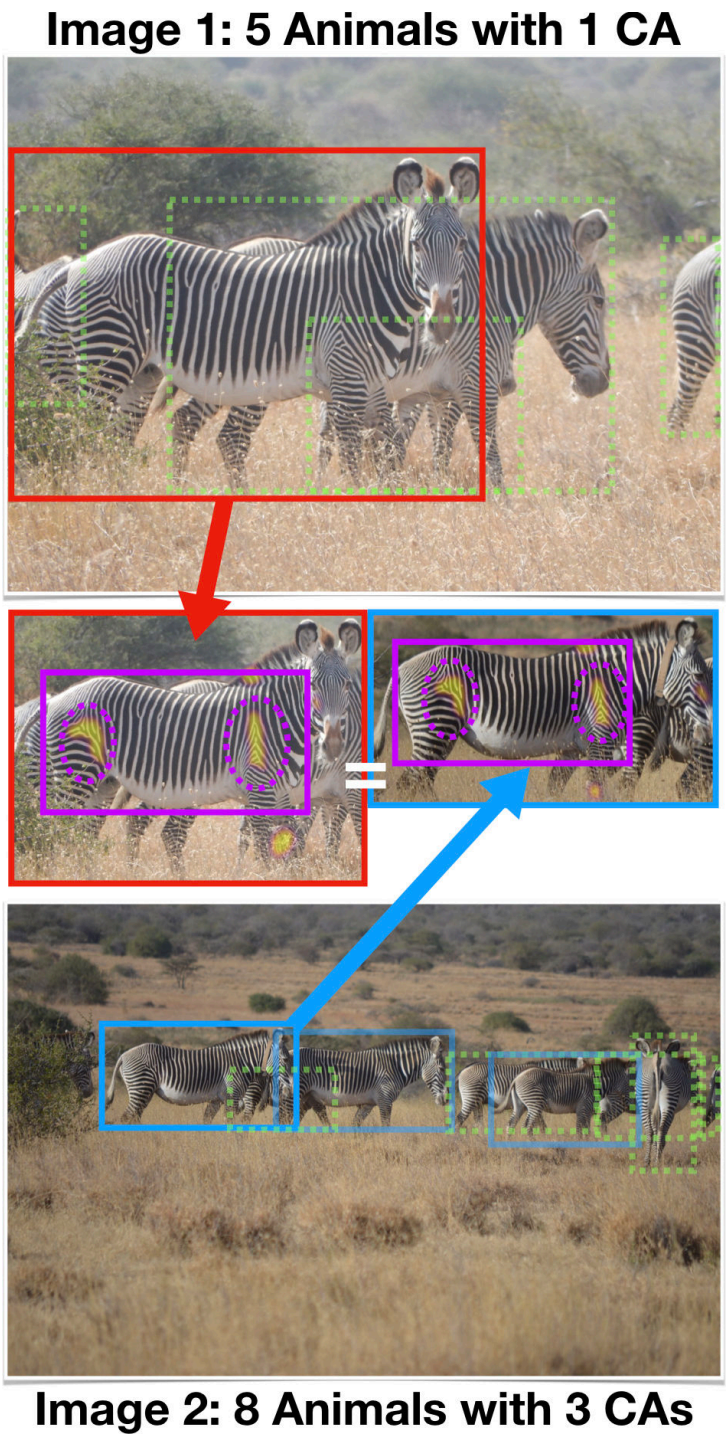


Figure 5.1: **An example of identification matching with two Census Annotations. Image 1 (top) shows a Census Annotation (CA) in red, which captures the same individual as the blue Census Annotation in Image 2 (bottom). The matched CAs (purple boxes) both contain the distinct chevron and hip regions (ovals) that commonly match by ID.**

5.1 Census Annotation Dataset

Both the Census Annotation and Census Annotation Region components were trained on a dedicated dataset for the problem and are evaluated as stand-alone machine learning approaches, separating them from the photographic censusing experiments later in this chapter and the creation of the GZCD from Chapter 4. A collection of 10,229 Grévy's zebra and 2,322 reticulated giraffe annotations were exported from the Great Grévy's Rally 2018 dataset, along with their original images and any other annotations that were also seen in those images. A reviewer was provided with a grid of annotations in a web interface to select the annotations that were CAs for the respective species. For this task, a CA was defined as any annotation that clearly showed (i.e., in focus, well lit) a right-side Grévy's zebra with its hip and shoulder region visible. For a reticulated giraffe, the body region needed to be fully visible (ignoring any ability to see the neck, head, and legs) for its annotation to be considered a valid CA. Each grid presented 500 annotations, and the reviewer toggled the state for a specific annotation by clicking on an image's thumbnail. The decisions were saved in bulk for the entire grid, and the process was repeated with subsequent grids of unlabeled annotations until the entire dataset was reviewed. Once all ground-truth CA decisions were added, a second independent review (by a second person) was performed for all negative "non-CA" (NCA) annotations and positive CA annotations, as two separate groups, to cross-check for ground-truth errors.

In total, the manual ground-truth labeling resulted in 1,837 Census Annotations (17.95%) for Grévy's zebra and 230 (9.9%) for reticulated giraffe. Figure 5.2 shows examples of Census Annotations (right, no border) and non-CAs (left, red border) for Grévy's zebra (top two rows) and reticulated giraffe (bottom two rows). All of the Census Annotations for Grévy's zebra were then reviewed further with a different web interface (see Figure 5.3) to add Census Annotation Region bounding boxes. The CA-R bounding box (green box) was required to be axis-aligned with the CA's bounding box (red box) and was not allowed to extend outside the bounds of the original CA annotation's bounding box. In total, 1,837 boxes were annotated, one CA-R for each Grévy's zebra CA in the training dataset. The reticulated giraffe CAs were not annotated with CA-R boxes because of the relatively low number of examples for training and validation, with less than 50 annotations reserved for held-out experiments. The images in the dataset were then partitioned into separate train (80%) and validation (20%) sets and stratified such that a balanced number of annotations per image occurred in each set.

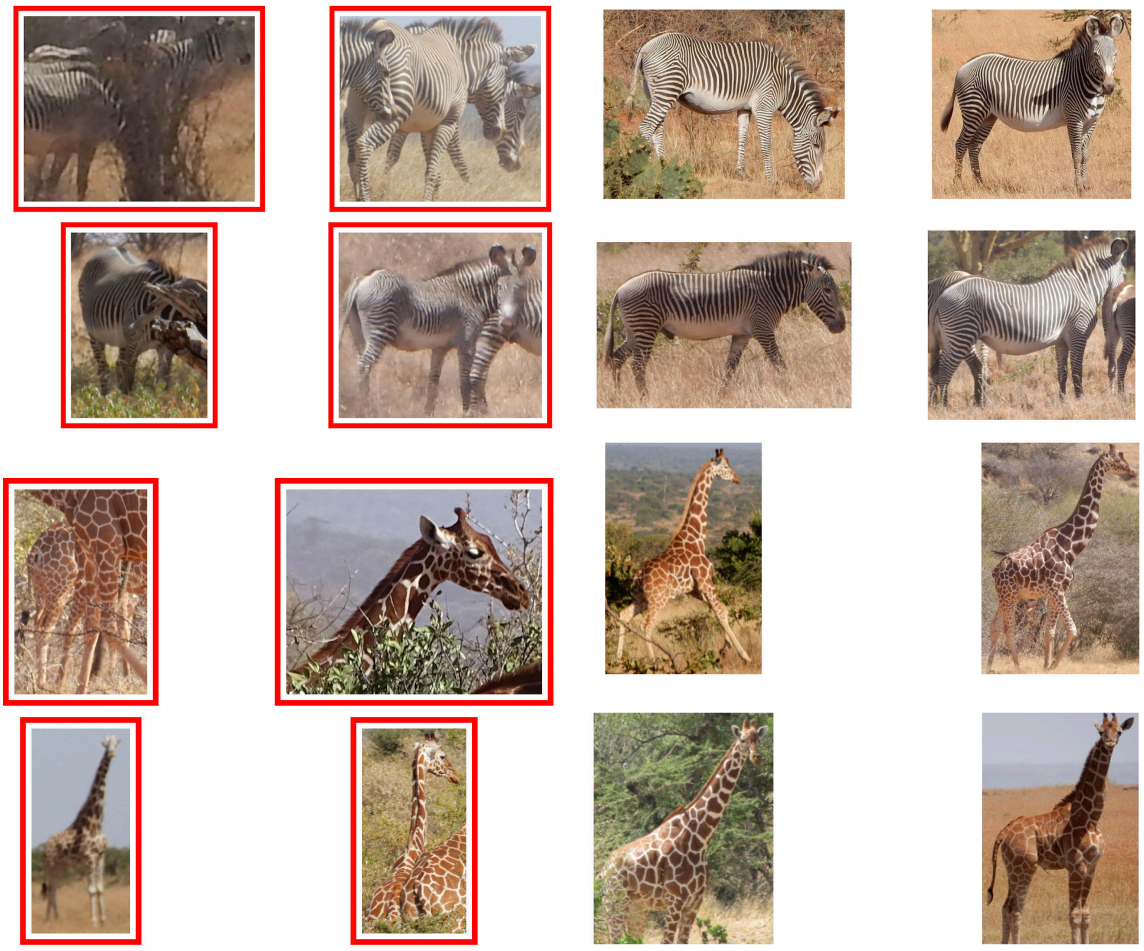


Figure 5.2: Example images of Census Annotations for Grévy's zebra and reticulated giraffe. Annotations that were marked as non-CAs by a reviewer (left two columns) vs. marked as census (right two columns). Grévy's Zebra are displayed on the top two rows and reticulated giraffe are shown on the bottom two rows.

5.1.1 Comparison to Annotation of Interest (AoI) and Quality

The concept of Census Annotation can be viewed as the annotation-level complement to Annotation of Interest (AoI). The notion of an "annotation of interest" is inherently an image-level determination on the primary subject(s) of an image and functions to determine which animals were incidentally seen in the background. This focus on finding good quality foreground sightings of animals is related to Census Annotation, but AoI and CA do not overlap perfectly in their goals. In contrast, Census Annotation represents the comparability between two annotations. The notion of CA ensures that reliable identifying information is always available, regardless of where and how the annotation exists in the image composition. Figure 5.4 provides examples of when AoI and CA

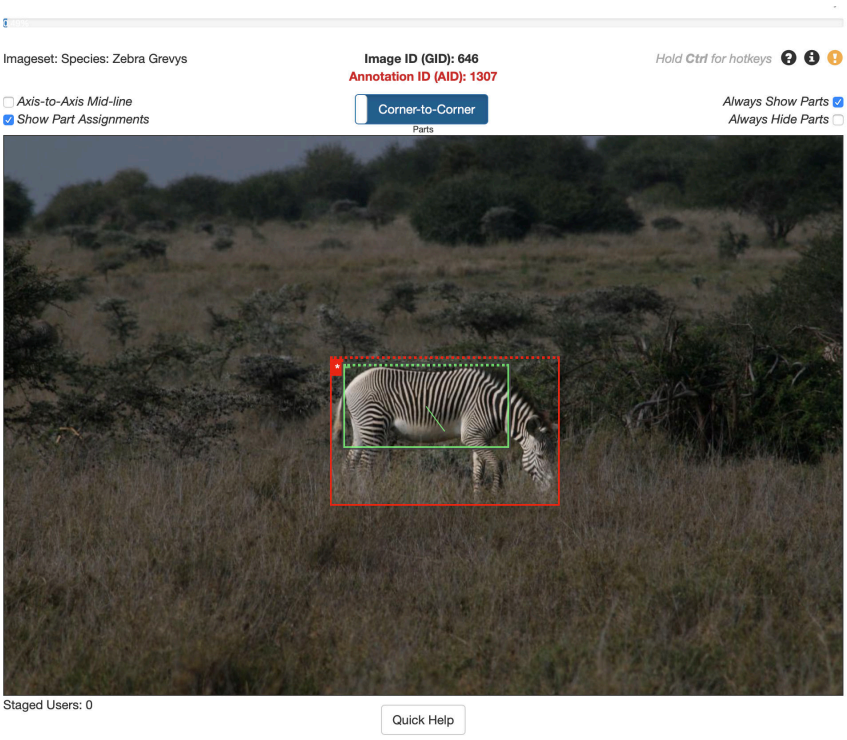


Figure 5.3: An image of the Census Annotation Region web annotation interface. The web interface used to annotate Census Annotation Regions (green box) onto existing Census Annotations (red box). CA Regions are assigned to an existing annotation as a "part" and can inherit important metadata like species, viewpoint, and name assignments.

can disagree. While an annotation that is not an AoI is also not likely to be a Census Annotation, it is not guaranteed to be the case, as seen in the bottom row. An annotation can also be considered an AoI based on the semantic context of the image but does not show enough clear and reliable ID information to be considered a Census Annotation (top row).

Furthermore, the detection pipeline's labeler has the ability to predict an explicit "quality" value for annotations (e.g. *junk*, *good*, *perfect*). This feature was originally used in previous experiments with photographic censusing (see [2]) to filter annotations for ID curation but was ultimately abandoned. The problem is that "quality" is a fairly subjective measurement (i.e., it is hard to get consistent ground-truth labels) and does not guarantee that two acceptable quality annotations will be comparable. Although a quality metric and AoI can be highly correlated with Census Annotation, they are insufficient replacements when the goal is to eliminate incomparable decisions from ID curation. As such, the eventual formulation of Census Annotation is the culmination of real-world experimentation with large censusing events that have failed to achieve high degrees of

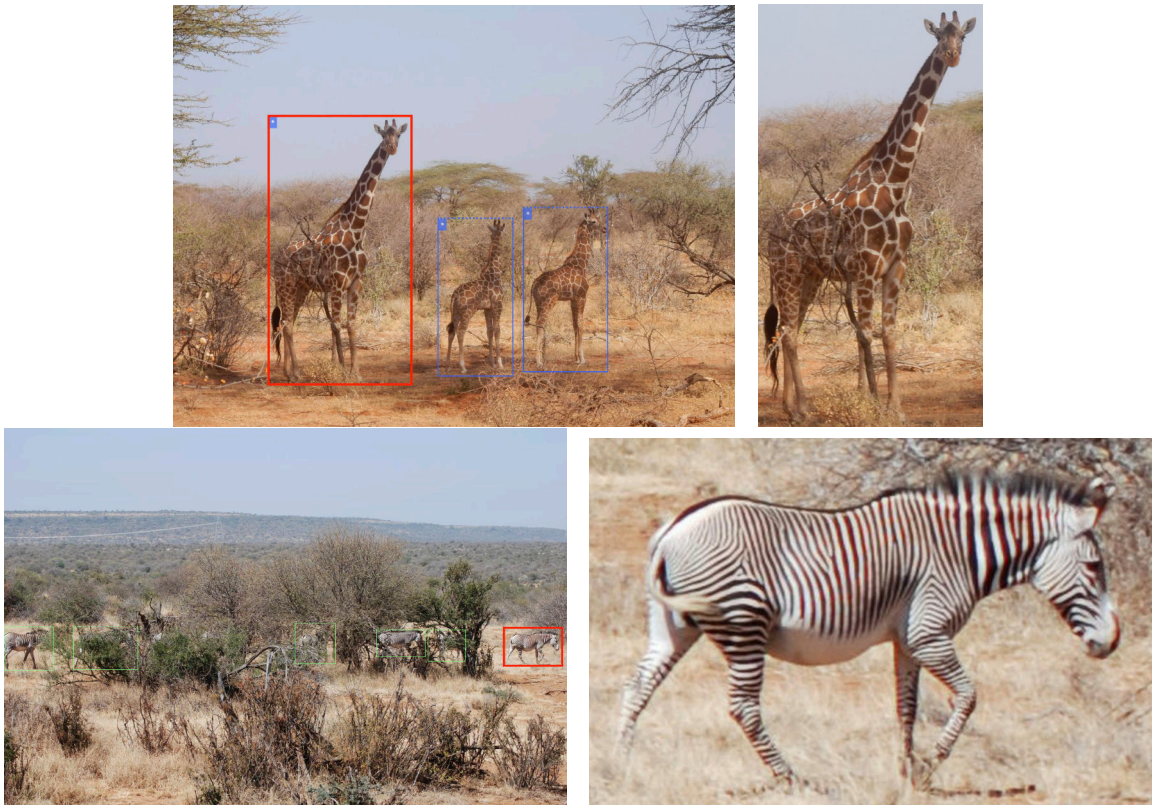


Figure 5.4: Example images of the disagreement between AoI and Census Annotation. The top row provides an example of an annotation that is an AoI but not a CA, with the annotation (left) and image (right). The red annotations in the images are provided at a higher resolution to the right. The giraffe is a borderline AoI due to its occlusion, but it is one of the primary subjects of the image and is decidedly in the foreground. The bottom row gives an example of a Census Annotation that is not an AoI. The animal is clearly comparable as seen by the annotation but is seen far away (small scale) and is a member of a herd, two items that make it a difficult case for AoI.

automation (due to challenges discussed previously in Chapter 4).

The notions of AoI, quality, and CA can be compared and contrasted using the ground-truth labels in the GZCD. In that dataset, there is a total of 9,205 Grévy's zebra annotations. Of those, 4,246 have been ground-truth annotated as AoIs, 7,372 show some degree of the animal's right side, and 4,119 have an acceptable quality value. In addition, there are 4,005 Census Annotations in that dataset, with a 93.4% overlap with a quality filter and 87.4% overlap with AoI. In summary, these values suggest that a quality metric and even AoI can be somewhat helpful in filtering annotations, but with the addition of CA to the detection pipeline, they are largely redundant. That being said, Annotation of Interest is still used to validate the detector's performance and is still worthwhile to

Figure 5.5: Examples of on-the-fly training augmentations for the Census Annotation classifier. Positive samples are highlighted with a green border (CA examples) while negative samples have a red border (Non-CA examples). Each example received a unique, randomized augmentation for each epoch and was computed on-the-fly.

annotate, but a quality value for each annotation is no longer necessary as Census Annotation has superseded it. We will now shift our attention to the implementation details of the CA and CA-R methodologies and analyze their stand-alone performances as new detection components.

5.2 Census Annotation (CA)

The Census Annotation classifier was trained using a pre-trained DenseNet 201 [89] feature extraction network with a linear classification layer added on top. The network was fine-tuned with SGD (LR 0.001, momentum 0.9, and a ten epoch patience LR schedule) and used a standard Cross-Entropy loss. The input images were sent through a moderate level of data augmentation (e.g., contrast normalization, per-channel pixel noise, hue and saturation changes, piece-wise Affine

transformations on a small grid, and slight rotations and shearing). Example augmentations for a given image (top row) can be seen in Figure 5.5 for negative non-CA (red border) and positive CA (green border) annotations. Furthermore, each mini-batch was sampled such that it had a balanced number of positive and negative examples. The classifier was trained as an ensemble of three separate neural network models (each with unique initialization), and their respective outputs are averaged into a single prediction during inference. This neural network design is standardized and shares principles with existing detection pipeline classification components like the annotation labeler (see Section 3.4). The CA component remains independent of previous detection methods, however, and its modular implementation can be disabled or updated as needed without impacting other components.

A series of four CA classifier models were trained using different configurations of data augmentation and iteratively better ground-truth training data between each version. The training examples for each CA classifier were selected using a simple species filter and did not consider viewpoint or quality. The purpose was to train the CA models on ideal CA examples, annotations that were obviously incorrect (e.g., wrong viewpoint), and annotations that had relatively good quality overall but were ultimately incomparable. The final version (V4) has the best performance on the held-out validation data (see Figure 5.6, left). This result is expected as the data augmentation scheme was improved for that model to be more aggressive (acting as a better regularizer). In addition, it was trained on the cleanest data after label corrections were applied to the ground-truth. In total, 26 CA ground-truth labeling errors (0.6%) were identified in the GZCD and fixed by hand. Overall, the V4 model achieves a classification accuracy of 96.8% for Grévy's zebra using an operating point (OP) of 0.31. In addition, the model makes 54 false positive (FP) decisions compared to 7 false negatives (FN). This balance of errors is a good trade-off for a filtering component because we can treat Type I misclassifications as extra work (and not invalid data) during ID curation. Therefore, the practical success rate of the classifier is 99.6% when we consider false negatives are the most problematic source of error.

The Census Annotation classifier was also trained and evaluated on reticulated giraffes. Even though the total number of annotations was significantly smaller, the CA classifier still did well to classify 91.3% of the examples correctly (see Figure 5.6, right). Suppose we apply the same logic about false positives being a less worrisome type of error. In that case, the model only makes two mistakes out of 470 annotations (99.6%) on held-out validation data for the price of reviewing an additional 39 annotations. Upon inspection, the annotations that the zebra and giraffe

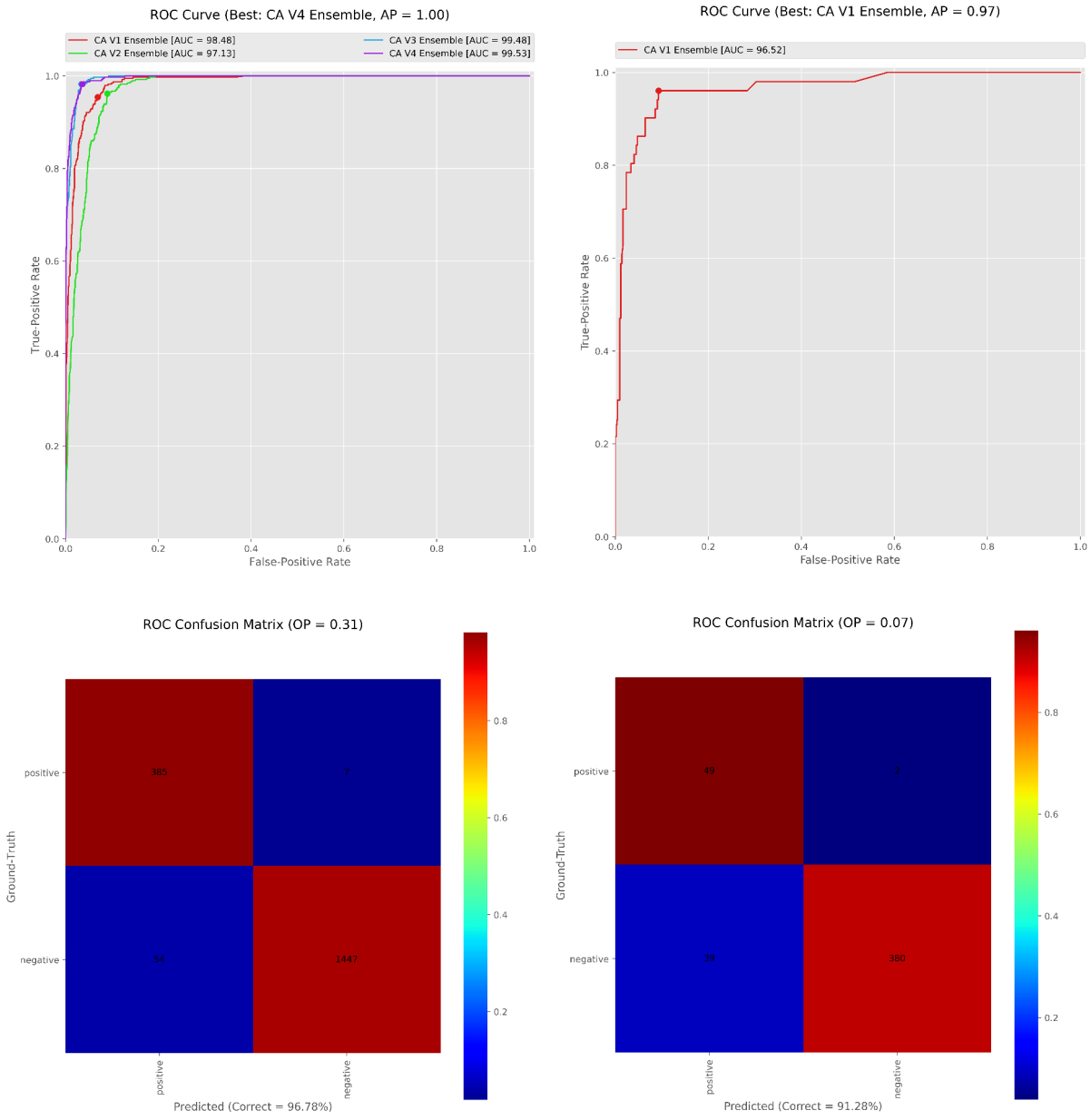


Figure 5.6: The ROC performance curves for the Census Annotation classifier. Top: ROC curves showing the classification performance and their respective Area-Under-Curve volumes (AUC) for Grévy's zebra (left) and reticulated giraffe (right). Bottom: The Confusion Matrix for the best model (V4 for zebras, V1 for giraffe) and best operating point (zebra OP=0.31, giraffe OP=0.07) as determined by the colored dot in the various ROC curves. The accuracy of the Grévy's zebra CA classifier is 96.8% and 91.3% for reticulated giraffes, but if false positives are treated as extra work and not errors then the accuracy increases to 99.6% for both models.

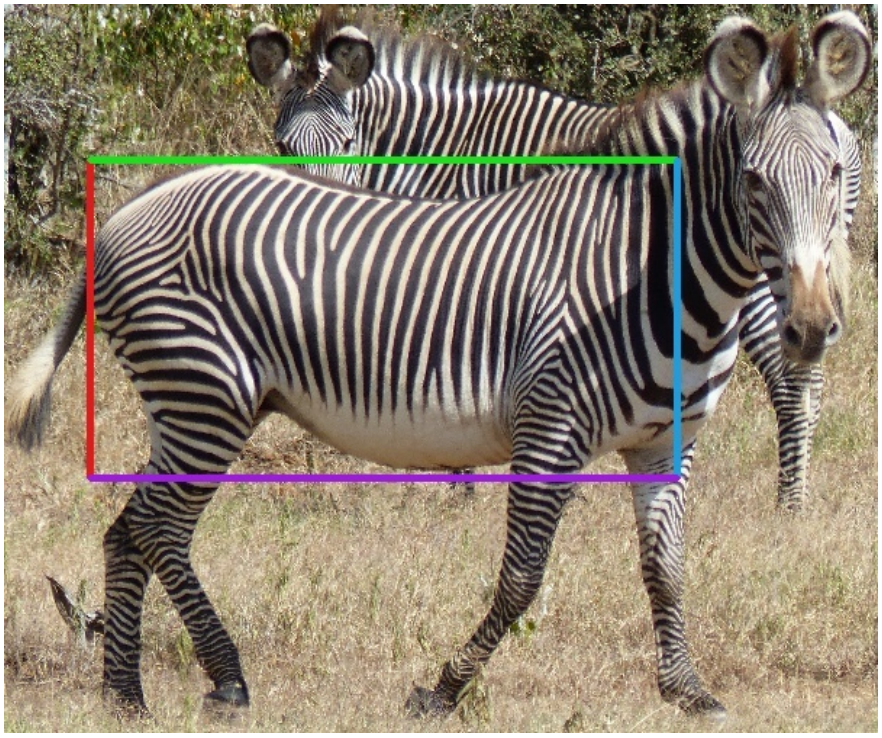


Figure 5.7: An example image of a Census Annotation Region, which is defined by its four edge components: `x0` (left, red), `x1` (right, blue), `y0` (top, green), and `y1` (bottom, purple).

models incorrectly predict as CAs are mostly borderline (subjective) cases in the ground-truth labels. Ultimately, the number of giraffe annotations in the GZCD was too small for a worthwhile evaluation with CA-R. Moreover, the GZCD focused exclusively on Grévy's zebra, so a large, authoritative ID dataset does not currently exist to demonstrate any potential improvements in incidental matching. The lack of a reliable ID database for reticulated giraffes is primarily due to limited resources considering how much time and energy was spent curating and verifying the Grévy's zebra IDs in the GZCD. As such, the remaining analysis on CA-R, and following discussions in this chapter, are focused entirely on Grévy's zebras.

5.3 Census Annotation Region (CA-R)

The Census Annotation Region model was trained as a regression network that is tasked with predicting four simple values (shown in Figure 5.7): `x0` (left, red), `x1` (right, blue), `y0` (top, green), and `y1` (bottom, purple). All of the original CA bounding boxes are assumed to be rotated where the animal's head is at the top of the box. Furthermore, the CA Region bounding boxes were all

assumed to have inherited the rotation of their associated CA. This setup means that each of the four edges of the CA Region bounding box (top, bottom, left, and right) can be represented by a single value for how many pixels it is away from the corresponding edge of the CA's bounding box. All x-axis pixel offsets are converted to a decimal value from 0.0 to 1.0 by dividing by the annotation's width. This process is repeated for y-axis pixel offsets with the pixel height of the annotation. Therefore, the network was trained to predict a positive value between 0.0 and 1.0 for each of the four edges of the CA Region's bounding box as a residual from the original box's bounding box location.

The CA-R regression network has a similar convolutional back-end to the CA classifier: the model is fine-tuned with SGD (initial LR 0.0005), uses similar data augmentation as the CA model (but without translation or rotation operations), has a DenseNet 201 architecture with pre-trained weights, and uses a 4-node linear layer for its output. The network is optimized using a modified loss function for mean squared error (L2), with additional terms for the "over-shooting" (α) and "under-shooting" (β) components of the standardized regression loss. In general, the expectation is that any distracting background information is along the edge of the CA's original bounding box. We are trying to minimize background information, but that goal should not come at the cost of accidentally removing useful ID information on the animal's body. The resulting loss term is defined as:

$$\text{Loss}_{\text{CA-R}} = \sum_{n=1}^{n} \sum_{c=1}^{4} \alpha * (\min(0, x_{n,c} - \bar{x}_{n,c}))^2 + \beta * (\min(0, \bar{x}_{n,c} - x_{n,c}))^2 \tag{5.1}$$

where $x_{n,c}$ is the ground-truth value, $\bar{x}_{n,c}$ is the predicted value by the network, and c represents the four possible axes. The CA-R regression models are also trained as an ensemble of three separate neural networks, and their final results are averaged during inference.

A series of six CA-R ensembles were trained, partially to help identify and correct ground-truth errors similar to the process used for the CA classifier. The first three models (versions 1, 2, and 3) were used to bootstrap a better, cleaner dataset and were discarded after the issues they identified were fixed. The next three models (versions 4, 5, and 6) are useful to compare as they are trained on the same underlying CA-R data. When α and β are both set to 1.0, the network's loss is exactly L2 and is expected to balance the errors from over-shooting against under-shooting equally. This loss formulation is a problem, however, because it suggests that half of the predicted boxes will

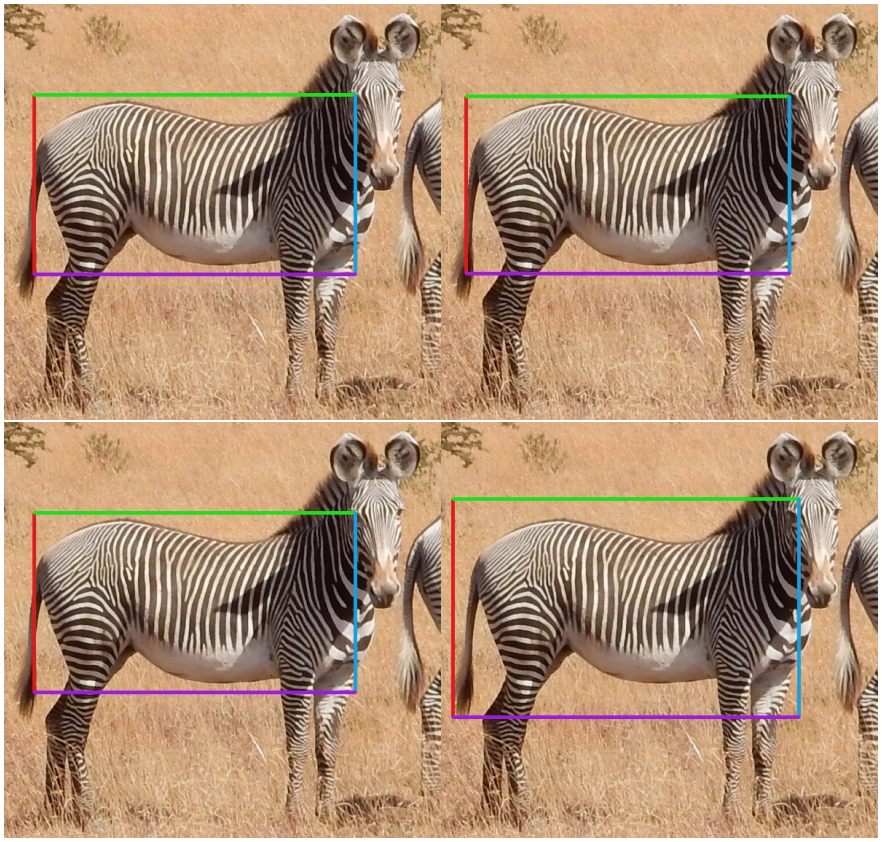


Figure 5.8: An example comparison of a Census Annotation Region output with different training configurations for overshooting. The figure shows an example output (right column) for the CA Region model V6 (top row) vs. V4 (bottom row) for an input annotation (left column). The input annotations to both models are identical. Two networks are offered to precisely predict the box on the margin (V6) or prevent overshooting (V4). We should prefer the larger predicted CA Region because it has less of a chance of throwing away useful information for ID.

have edges that overshoot the target offset (making the CA-R box too small, cropping out useful ID information) and the other half will undershoot (making the box too large, increasing the likelihood of incidental matching). Overall, the ideal CA-R model should be trained to eliminate as much under-shooting as possible while doing very little (if any) over-shooting. The V4 model was trained with a 4:1 ratio (quadruple the loss penalty for over-shooting, $\alpha = 4$, $\beta = 1$), V5 with a ratio of 2:1 ($\alpha = 2$, $\beta = 1$), and the V6 model with a 1:1 ratio (L2 norm, $\alpha = 1$, $\beta = 1$). By comparing the relative errors of each model configuration, the preferred behavior can be selected when creating CA-R bounding boxes. Figure 5.8 shows an example of what kinds of predicted boxes the V6 and V4 models generate.

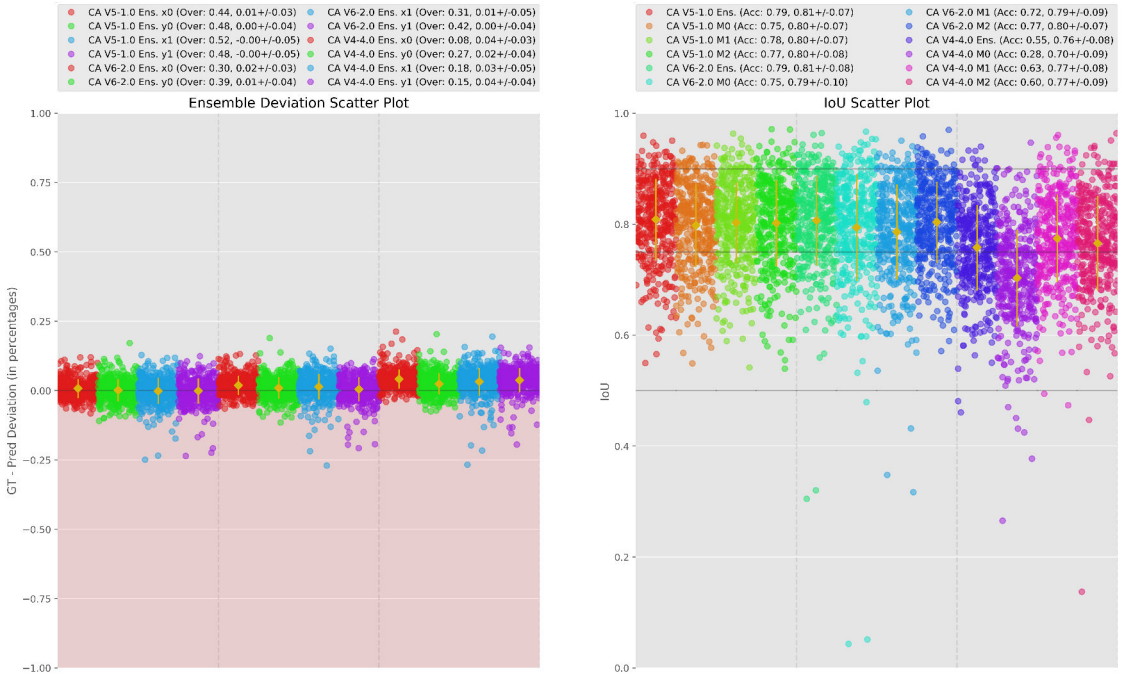


Figure 5.9: The regression performance curves for the Census Annotation Region component. Left: A deviation plot of each of the four edges for each of the three models. Right: An Intersection-Over-Union (IoU) scatter plot showing how well the predicted CA Region overlaps with the ground-truth box. An IoU greater than 0.5 is generally considered a correct detection with a value of 0.75 has a high degree of overlap.

Figure 5.9 (left) shows that the 3 models do a very good job at approximating the values for `x0`, `x1`, `y0`, and `y1`. In this plot, the values that are less than 0 on the y-axis are highlighted (in red), as this indicates a overshoot of the prediction. While the V6 predictions are mean centered at 0.0 (or 1% for `x0`), approximately half of its predictions are overshooting the ground-truth location (`x0` 44%, `x1` 48%, `y0` 52%, and `y1` 48%) as expected. This is something we want to aggressively avoid because it may remove useful information from ID curation. Model V5 was trained with twice the penalty term for overshooting and, while its center predictions are a bit worse, it does a better job at controlling overshooting (`x0` 30%, `x1` 39%, `y0` 31%, and `y1` 42%). The v4 model does the worst job at predicting the margin values (off by 2 to 4%) but does an excellent job at preventing over-shooting (`x0` 4%, `x1` 2%, `y0` 3%, and `y1` 4%).

Each CA Region model was trained as an ensemble of 3 separate neural networks, each with different initializations. The scatter plot in Figure 5.9 (right) shows the Intersection over Union (IoU) of the predicted boxes with the target ground-truth CA-R boxes. For detection tasks, the IoU threshold for a true positive (TP) detection is often specified as 50%. However, the CA Region

bounding boxes have a more vital need for precision, so the IoU threshold was required to be at least 75%. For each model, the percentage of predictions above this IoU threshold is reported as its accuracy. The performance of the ensemble for each model (e.g., `CA V5-1.0 Ens.`) is plotted next to the individual performance of their respective ensemble members (e.g., `CA V4-4.0 M0` to represent "Model index 0 within the V4 ensemble"). The V5 ensemble has one of the best accuracies at 79% and reports a better performance than its component models (V5 M0, V5 M1, or V5 M2). The ensemble's averaging effect also benefits the performance of the V6 model, where it also reports an IoU accuracy of 79%. Comparing it against the V5 model, the V6 model provides almost identical IoU accuracy but overshoots the box size significantly less often (a reduction of 14%). These results contrast with V4, which objectively produces the worst IoU accuracy (55%) between all models. Luckily, this drop in performance is not all that unexpected since the model was explicitly asked to predict less accurate boxes by design. A noticeable outlier is V4 Model 0, which scores a considerably lower IoU accuracy of only 28% and offers the lowest average IoU across all trained models. This poor performance is most likely due to a poor initialization state since Models 1 and 2 for that ensemble performed considerably better. Even with the worse M0 model, the V4 ensemble has an average IoU of 76%, above the required target threshold.

Since the V4 model still has reasonably good bounding box prediction (only a handful of examples score below 50% IoU) and the amount of overshooting is considerably lower for all four axes, it is used for the remainder of the experiments where CA-Rs are required. An immediate question now that we can create CA-Rs automatically is, "*how much easier are they for humans to verify?*" This question, when restricted to comparable pairs of annotations, is fundamentally also a question about how much time it takes for a human to verify a match.

5.4 User Study on Human Speed and Accuracy

We need to determine if using Census Annotations in a photographic census positively impacts a human's ability to decide the matched pairs sent to review. Specifically, we need to know if CAs or CA Regions 1) improve the accuracy of manual verification and 2) reduce the total amount of time it takes to make a decision. With an automated population census, the primary goal is to minimize the number of verification decisions needed from humans. When human work is needed, though, there is a secondary goal of minimizing the complexity of the verification task; this is important because we can expect that an easier decision will end up being faster, thus reducing the total on-task time for humans. The GZCD provides a ground-truth dataset of hundreds of animal IDs, with some

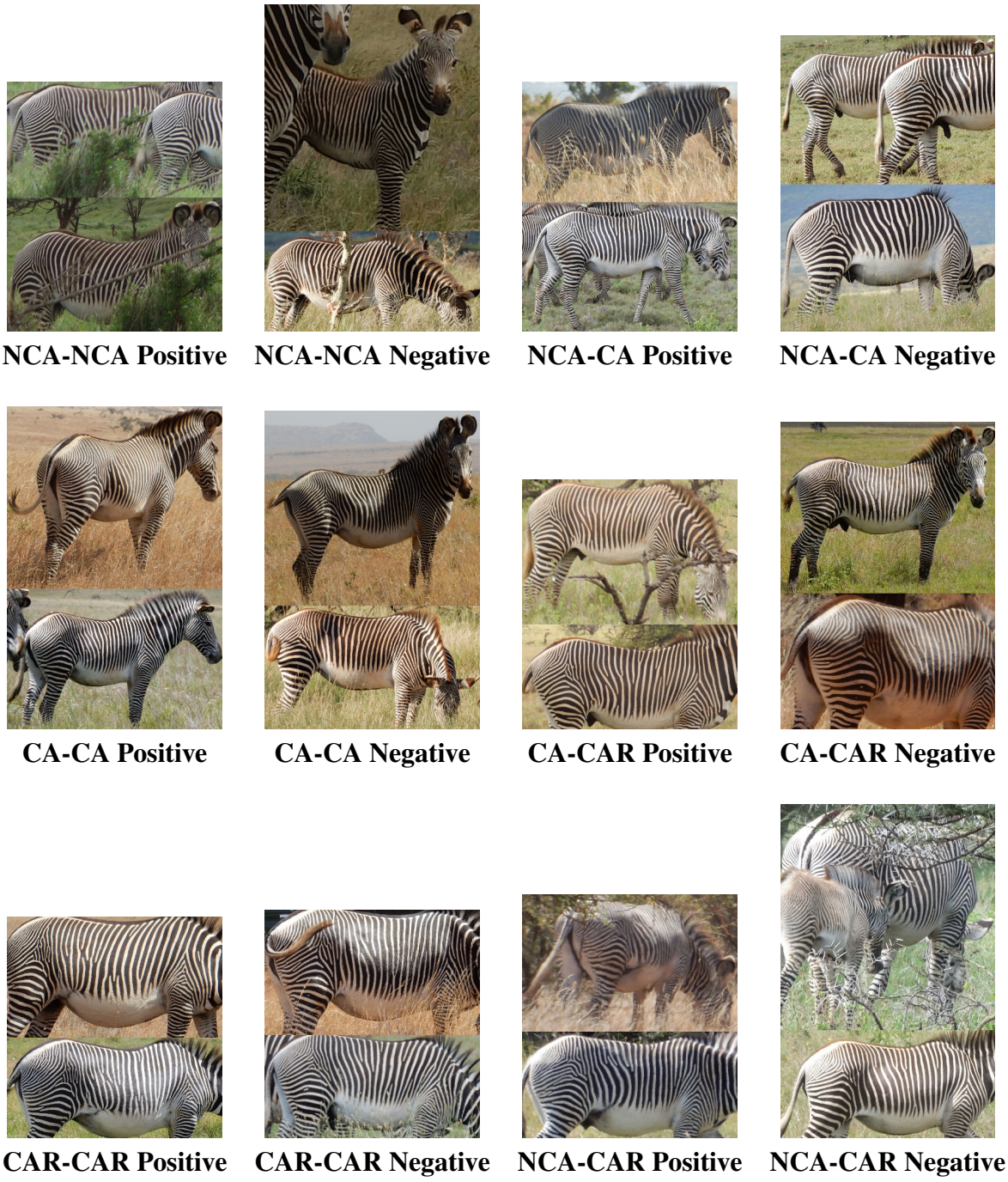


Figure 5.10: Example match pairs used during the user study. The user study was designed to test the impact of Census Annotation and Census Annotation Regions on human verification, measuring the accuracy and time it took to review 300 total pairs. The expectation is that a reviewer will have the most difficulty (and therefore spend the most time) with NCA-NCA pairs and perform the best with CAR-CAR pairs.

of them containing relatively poor-quality annotations. The dataset is mined for pairs that range from being easy to review (a pair between two Census Annotation Regions) to hard to review (a matched pair between two non-CAs) to evaluate the performance of humans on reviewing pairs. For simplicity, this section will refer to any annotation that is not a CA as a "non-CA" or "NCA". Furthermore, a Census Annotation will be referred to as a "CA", like usual, and a CA Regions as a "CAR" (without the ordinary hyphen). Match pairs between two annotations will be (conveniently) denoted with a hyphen. For example, a "CA-CAR" pair contains one Census Annotation and one Census Annotation Region, and the ordering does not matter. Having three possible states for an annotation produces a combination of 6 possible pair types to collect for this user study:

- **NCA-NCA** - a non-CA and non-CA pair
- **NCA-CA** - a non-CA and CA pair
- **NCA-CAR** - a non-CA and CA Region pair
- **CA-CA** - a CA and CA pair
- **CA-CAR** - a CA and CA Region pair
- **CAR-CAR** - a CA Region and CA Region pair

For each of the six types of pairs, there are ground-truth "same animal" (positive) examples or "different animals" (negative) examples, resulting in 12 total options for the study. To properly sample the various combinations of pair types, all of the named annotations (10,037 total, 4,762 named CA Regions) in the GZCD (that also had an acceptable quality) were selected. This filtering resulted in a set of 9,966 annotations and CA Regions to mine for match pairs: 1,692 NCAs, 4,142 CAs (as determined by a threshold of 0.31), and 4,142 corresponding CA-Rs. All of the $\binom{9966}{2}$ pair combinations (49.6 million in total) were enumerated and were randomly shuffled. The random collection was then traversed, and pairs were gathered for each of the 12 categories. The pair's category was determined by 1) the ground-truth NCA, CA, or CA-R status of its annotations and 2) a "same animal" or "different animal" decision from the ground-truth name IDs between its annotations. The process continued until 25 examples for each category were found, generating a total of 300 pairs. An additional constraint on the mining process required all 600 annotations (two per pair) to be unique. Figure 5.10 shows an example for each of the 12 types. A final check was performed by hand to ensure that none of the 300 pairs were accidentally incomparable. Each pair was guaranteed to be decidable given enough time and subject to the expertise of the individual user in the study.

The collection of 300 test pairs was then given to six independent reviewers. Three reviewers

in the study are considered "experts" in reviewing pairs of Grévy's zebra matches, each with the real-world experience of reviewing thousands of match decisions. Three additional "novice" reviewers were also added to the study – two with zero experience with the problem domain and task – and functioned as a way to control for task experience. A web-based interface was created that allows each user to annotate a "same animal" or "different animal" decision for each pair, one at a time. The web application measured the total turn-around decision time and recorded the accuracy of each decision. The time between decisions was not tracked (i.e., a reviewer had to request the next match manually) and was allowed to take breaks as needed. The decisions were made blind without indicating the correct decision and without any algorithmic hints or highlighting where the two annotations may have matched. The users were provided with a brief training session of approximately ten pairs to understand the task and interface and were told that exactly 50% of the matches were "same animal," and 50% were "different animals". The ordering of the match pairs was randomized between each user, and the user was instructed to make decisions at the fastest pace possible while also making as few errors as possible.

Between the six participants in the study, some reviewers were very fast (6.1 seconds per decision) while others were slower (15.9 seconds). The web interface was optimized with pre-rendered images. Each user's decisions were collected at non-overlapping times, and the webserver used the same hardware setup for all participants. What was not controlled for was the total round-trip travel time of the loaded web content over the Internet or the various computers, browsers, and medium that each participant used. For example, one participant was located in the same city as the webserver, while another was located three time zones away and in a different country at the time of their participation. However, a constant delay experienced by a given user during the study is expected to be somewhat reliable. All users completed their participation in the study in less than 90 total active minutes and freely volunteered their time without financial compensation. For transparency, the author of this dissertation and his Ph.D. advisor both participated as "expert" users.

A user's average decision time for all 300 decisions is subtracted from the time for each decision to counteract the effects of any communication delay. This correction results in a global mean of 0.0, where faster-than-average annotation pairs have a negative seconds time value and slow pairs have positive time values. However, additional correction is needed because users became more comfortable with the task as the study progressed. The users, especially the novice users, were faster in their decision-making towards the end of their participation in the study than at the

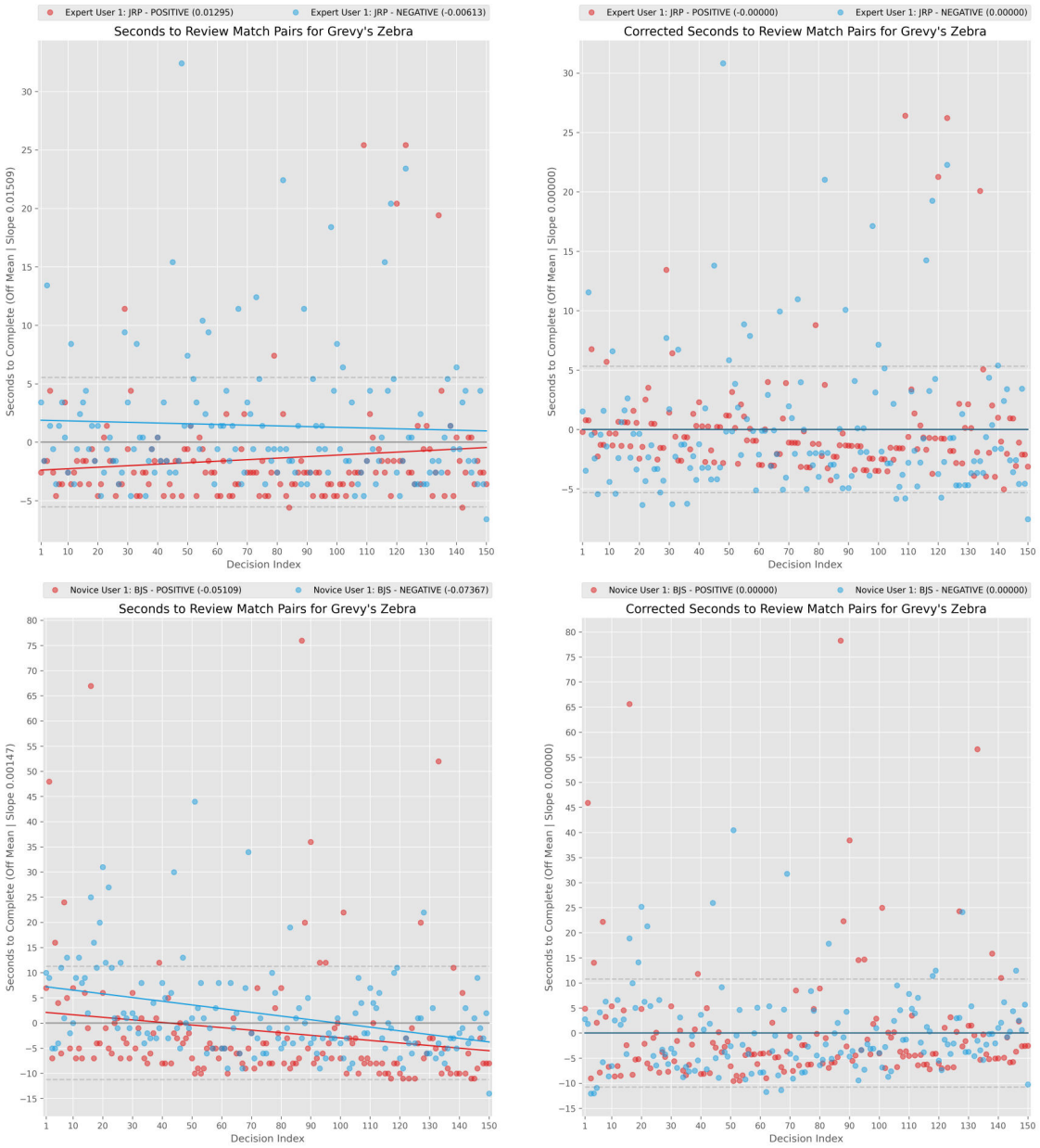


Figure 5.11: The decision times for various match pairs as seen during the user study. The "off mean" times to complete 150 positive "same animal" and 150 negative "different animal" match pairs (300 total). The time for an expert (top) and a novice (bottom) are shown, with the original times (left) and the slope-corrected times (right) displayed for both users. The positive slope for the expert's "same animal" decisions (red line) indicates that the user slowed down over time for those pairs. The novice user, in contrast, grew more comfortable with the study as it progressed and was faster for both "same animal" (negative blue line slope) and "different animals" decisions (negative red line slope).

start. Furthermore, it is generally easier (and therefore faster) to review negative "different animal" pairs because not as many points of comparison are needed to prove two animals are different. For example, once a user identifies a pattern that does not match, they quickly move on. To verify a positive match, users tend to be a bit more careful and look for more than one point of comparison, which takes additional time. For all of the participants – experts and novices alike – their 150 positive pair decisions were slower on average than the 150 negative pair decisions. Two lines were then fit to the "off mean" times for positive and negative examples to correct these effects. The slope of each line was then used to offset the individual times for each decision where the positive and negative mean times were equal to the global mean. The plots in Figure 5.11 show the original and corrected times for an expert (the author, top row) and a novice (bottom row). The expert was more consistent in their review times (standard deviation 5.5 seconds), while the novice, as expected, was more varied ($\sigma = 11.2$ seconds).

After each user's decision times are corrected with their own unique global mean and calculated slope offsets, the relative time spent reviewing NCA-NCA, CA-CA, and CAR-CAR pairs can be calculated and compared. Each user was shown 50 examples (25 positive, 25 negative) of each category during the user study, randomly interlaced with the other types and combinations of pairs. Figure 5.12 shows the off-mean times each user spent on each match pair type. We can see that NCA-NCA pairs were significantly slower for all users than their mean decision times. On average, each user spent 4.5 additional seconds on these pairs, indicating they were more challenging to review. On the other hand, match decisions between CA-CA pairs were much improved and were faster than the average by 0.8 seconds. Importantly, each user was as fast as their average decision time with CAs or was noticeably faster. Lastly, match pairs between two CARs were substantially faster, saving 3.0 seconds on average per decision. As for accuracy, the expert reviewers had an average accuracy of 98% compared to the novice users, who averaged at 94.1%, with the lowest individual accuracy of 91.7%.

Let us now consider a photographic census where only comparable annotations are used but no other restrictions on quality are applied. Let us also assume that any human review needed during ID curation is an unbiased collection across the 12 match pair types from above. The results from the user study suggest that the average rate of decisions for all users is approximately 400 decisions per hour per user and an accuracy rate of 96.1%. If we consider a photographic census constructed out of only Census Annotation Regions, the decision throughput increases to 560 decisions per hour, a 40% relative speed improvement. Furthermore, the study users made 71 decision errors, and

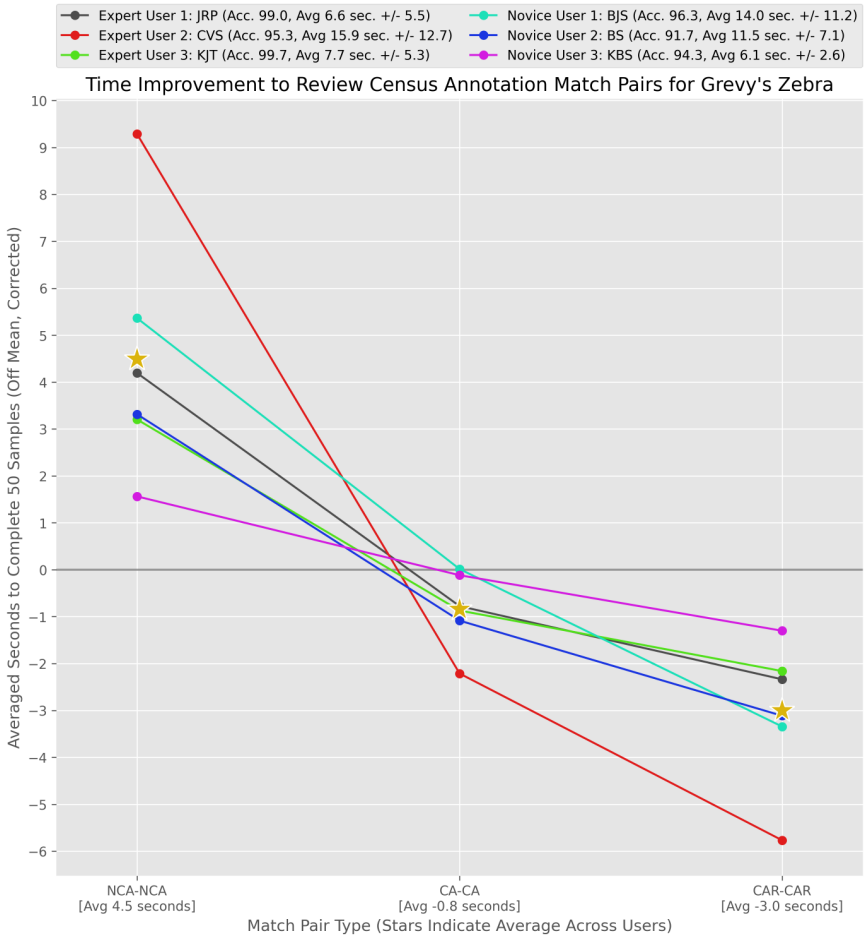


Figure 5.12: A comparison of the decision times for each match pair type. All six users were given 50 match examples between two non-CAs (NCAs), two Census Annotations (CAs), and two Census Annotation Regions (CARs). The slowest pairs to review were the non-CAs at 4.5 seconds (on average) slower than each user's unique mean. The fastest pairs to review were the CAR-CAR pairs, with an average time savings of 3.0 seconds per decision.

52 of those errors were with pairs that contained at least one NCA. If NCAs are excluded from ID curation, the number of errors made by the human verifiers would have been reduced by 73.2%. Lastly, we can compare the match pairs on which the study users spend the least amount of total (off mean) time against the pairs they spent the most combined time on. Figure 5.13 shows the five fastest and five slowest pairs, which clearly shows that CAs and CA Regions are a strong indicator for how easy and quickly annotation pairs can be reviewed by humans.

While the impact on human decision-making is vital to analyze, it is not the only photographic censusing process that decides match pairs. Automated verifiers are used extensively during ID

Figure 5.13: Example images of the fastest and slowest match pairs during the user study. Out of the 300 match pairs in the user study, the five annotations that users spent the most time on (slowest) and the five annotations that users spent the last time (fastest) are shown. We can see that the slowest match pairs to review have very hard to compare viewpoints and visual information that is obscured. The fastest annotations show clearly at least one of the two comparable regions for Grévy's zebra Census Annotations, with two of the fastest five matches being CAR-CAR pairs.

curation as the desired replacement for human reviewers, and the impact of CA and CA-R represents a substantial opportunity to improve automation. Now that CA-Rs have been shown to dramatically improve human verification performance and decision times, we need to analyze what improvements they may have with automated match verification.

5.5 Analysis on Separability of Automated Decisions

VAMP is a verification algorithm developed by Crall [13] that decides if pairs of annotations are one of three mutually-exclusive states: 1) "same animal" (*match*), 2) "different animals" (*no-*

match), or 3) "cannot tell" (*notcomp*; representing an incomparable decision). We would like to determine 1) how well the algorithm performs on held-out validation data with different types of annotation pairs and 2) how separable are the score predictions between positive and negative pairs. A collection of three VAMP models was created with the extensive ground-truth pair decisions provided by the GZCD. Each VAMP model is a set of cross-validated random forest (RF) classifiers trained on a fixed set of pairs mined from the animal ID database. The three VAMP models are defined by the annotations and match pairs on which they were trained. The following models were generated using all of the named annotations in the GZCD ID database (for a range of qualities):

1. Named Annotations - 20,046 pairs for 5,281 annots.
2. Census Annotations - 14,493 pairs for 4,142 annots.
3. CA Regions - 14,320 pairs for 4,142 annots.

Each model was validated for a fixed False-Positive Rate (FPR) of 1% and automated decision thresholds for `match` (same animal) and `nomatch` (different animals) were selected. Encouragingly, the "Named Annotations" VAMP model performed similarly to previous Grévy's zebra VAMP models trained on past censusing events for that species and used `match/nomatch` decision thresholds of 0.8062 and 0.8538, respectfully. These thresholds are used, for example, in the Graph ID algorithm to decide if a given annotation pair can be automatically reviewed. For comparison, a VAMP model trained on good-quality right-side Grévy's zebra annotations during the GGR-18 used thresholds of 0.7732 and 0.8605, respectively. In cross-validation, the model automatically decided 15,921 out of 20,046 of the pairs (79.4% automation). For `match` decisions, the model correctly decided 7,294 out of 8,717 pairs (84%) and 7884 out of 10460 `nomatch` decisions (75%). Since these annotations include non-CAs, the model predicts a third option for "notcomp" (an incomparable match). With a simple threshold of 50% (due to low relative volume of examples), the model automatically classified 355 out of 869 pairs (41% automated) but made 181 errors in the process (FPR 34%).

The CA and CA Region VAMP models were trained on fewer ground-truth pairs (14,493) than the "Named Annotations" model. However, this is an expected reduction as the total number of annotations decreased significantly from 5,281 to 4,142. The average number of reviews per CA is lower at 3.5 compared to the global mean of 3.8 reviews. The CA VAMP model was also configured to optimize for an FPR of 1% during cross-validation and has a much lower `match` automatic decision threshold. The model uses 0.6291 and 0.8695 for `match/nomatch` decisions and was

Table 5.1: The VAMP decision thresholds for three different sets of annotations. Three separate VAMP models were trained: 1) Named Annotations, 2) Census Annotations (CA), and 3) Census Annotation Regions (CA-R). The CA Region model performs the best and offers the highest degree of automation as it provides the cleanest version of each annotation for visual comparison.

Model	`match` **Threshold**	`nomatch` **Threshold**	**Automation**
Named [FPR]	0.8062	0.8538	79.4%
CA [FPR]	0.6291	0.8695	89.8%
CA-R [FPR]	0.5305	0.8496	94.6%
CA-R [MCC]	0.5000	0.5513	99.0%

able to automate 13,014 out of 14,493 decisions (89.8% automated) in total. This increased level of automation is not surprising as the VAMP model is only being given annotations that should be confidently decidable, as per the design of Census Annotations. The level of automation allows for 6,069 out of 6,540 `match` decisions to be automated (93%) and 6,801 out of 7,848 for `nomatch` (87%).

The CA-R VAMP model is expected to be the most discriminative at its task and, therefore, should achieve the highest levels of automation. No changes to the underlying name IDs or the number of total pairs for training were made compared to the previous CA VAMP model. The only change made was using the more focused bounding boxes around the identifying information for the Grévy's zebra and training VAMP on only those regions. The assumption here is that if a human reviewer decided a CA-CA annotation pair as "same animal", its associated CA-R to CA-R pair should inherit the same ground-truth decision. Unfortunately, this process could not work for all ground-truth human decisions, as 173 incomparable decisions could not be inherited. Any incomparable match decisions were intentionally left out of training as they represented only 1.2% of the match pairs. The resulting VAMP model for CA Regions, as expected, performs very well. Automated thresholds were selected for `match` and `nomatch` at 0.5305 and 0.8496, again using a FPR of 1%. This model allowed for automated decisions for 13,552 out of 14,320 pairs (94.6% automation). For `match` decisions, 6,236 out of 6,461 of the pairs were automatically decided, an outstanding level of automation at 97%. Likewise for `nomatch`, the model was able to decide 7,173 out of 7,753 pairs (93%). A summary of all three models, their decision thresholds, and their levels of automation can be seen in Table 5.1.

Furthermore, suppose the best CA Region VAMP model's thresholds are selected to optimize the classification task's Matthews Correlation Coefficient (MCC). In that case, the thresholds can be lowered even further to 0.5 (the lowest allowed) for `match` and 0.5513 for `nomatch`. These thresholds result in more errors overall and a higher level of automation as 14,182 out of 14,320 pairs are automatically decided (99.0%). To achieve this level of automation, 90 out of 6,340 (1.4%) `match` decisions are made incorrectly while 230 out of 7,842 (2.9%) `nomatch` decisions are incorrect (total FPR of 2.3%). For comparison, the "Named Annotation" model can also be tuned with the MCC to automate 96.9% of decisions but unfortunately has an overall FPR of 6.3% (5% for `match` and 7% for `nomatch`).

In summary, the ability of the CA Region VAMP model to quickly and accurately classify pairs reduces the need for human review during a photographic census. However, even with highly accurate automated verifiers, incidental matching can still be a problem for photographic censusing. The hope is that using Census Annotation Regions drastically reduces the number of inappropriate matches between annotations, which we will explore next.

5.6 Impact on Incidental Matching

As discussed in Chapter 4, the matching scenarios that are a significant source of error – and require human-in-the-loop review – are photobombs, mother-foal matches, and scenery matches. The problem is that automated matching between two annotations can sometimes be incorrect and match inappropriate visual information. Furthermore, incidental matching makes it difficult to accurately separate the rankings between true positives and negatives, hindering our ability to use decision thresholds. For example, scores for photobombs are hard to distinguish from correct matches because the ranking algorithm is correctly finding corresponding visual information. The solution to this problem is to remove the ability for distracting background information to match in the first place. Census Annotation Regions help mitigate the effects of incidental matching because it actively reduces the information within an annotation to only what is used to compare and contrast two annotations.

The GZCD contains many animal IDs and offers an extensive collection of match decisions made by hand; out of 43,048 human reviews, 1,540 were marked as explicitly showing a photobomb, and 1,007 pairs were labeled as a scenery match. Unfortunately, ground-truth match data on mother-foal photobombs were not collected during the ID curation of GZCD. However, this chapter's remaining discussion provides examples of mother-foal errors by the LCA algorithm and analyzes

the positive impact of using Census Annotation Regions for ID curation.

5.6.1 Photobombs

The GZCD ground-truth match decisions contain 1,540 photobomb pairs, with 239 pairs containing the same ground-truth ID (positive pairs) and 1,301 showing different animals (negative pairs). We would expect the vast majority of photobomb pairs to be negative decisions – indeed, it is 84.5% of the pairs – but it is not exclusively negative match decisions. For example, two annotations could have matched from a more distinctive background animal, but the primary animal in both annotations is the same individual. The candidate set of annotations was filtered to compare the Quality Baseline (defined in Section 5.7) set from previous sections to the set of CAs and their CA Regions. The Quality Baseline concerns 764 photobomb pairs, while the two CA algorithms filter out some of those to keep 592.

Figure 5.14 (left) shows the VAMP confidence scores for positive and negative ground-truth pairs. The scores are displayed and separated randomly on a scatter plot simply for easier viewing. It is clear from looking at the quality baseline for "different animals" that there is extreme confusion. The average VAMP score for these negative pairs is 43±26% and appears to be fairly uniform. Likewise, the Census Annotation scores are not better for "different animals" with practically the same average of 43±27%. However, for CA Regions, the VAMP scores are much lower with an average of 15±25% and are clustered closer together. The positive "same animal" examples improve slightly from 82±23% with the Quality Baseline to 89±18% but improve by over 10 points to 93±25% with Census Annotation Regions. By setting a decision threshold at 50% for VAMP, the Quality Baseline would classify 64.7% of the pairs correctly, CA with 63.5%, and CA Regions at 88.3% accurate. Picking an optimal operating threshold that maximizes all accuracy for each annotation set yields: Quality Baseline (OP 89%, accuracy 91.1%), CA (OP 94%, accuracy 92.1%), CA Region (OP 86%, accuracy 95.8%). These results indicate that photobomb cases are 1) significantly reduced in quantity by CA filtering and 2) substantially easier to classify correctly when Census Annotation Regions are compared.

A real-world example is provided in Figure 5.15. The original images (yellow border) with all annotations and the highlighted annotation (blue border) are matched visually. The annotations are both Census Annotations (red) but still result in a photobomb because the matched area (red circles) is likely to cause an incorrect "same animal" decision. Instead, when the Census Annotation Regions (green borders) are used, the photobombing annotation is removed from the bounding box

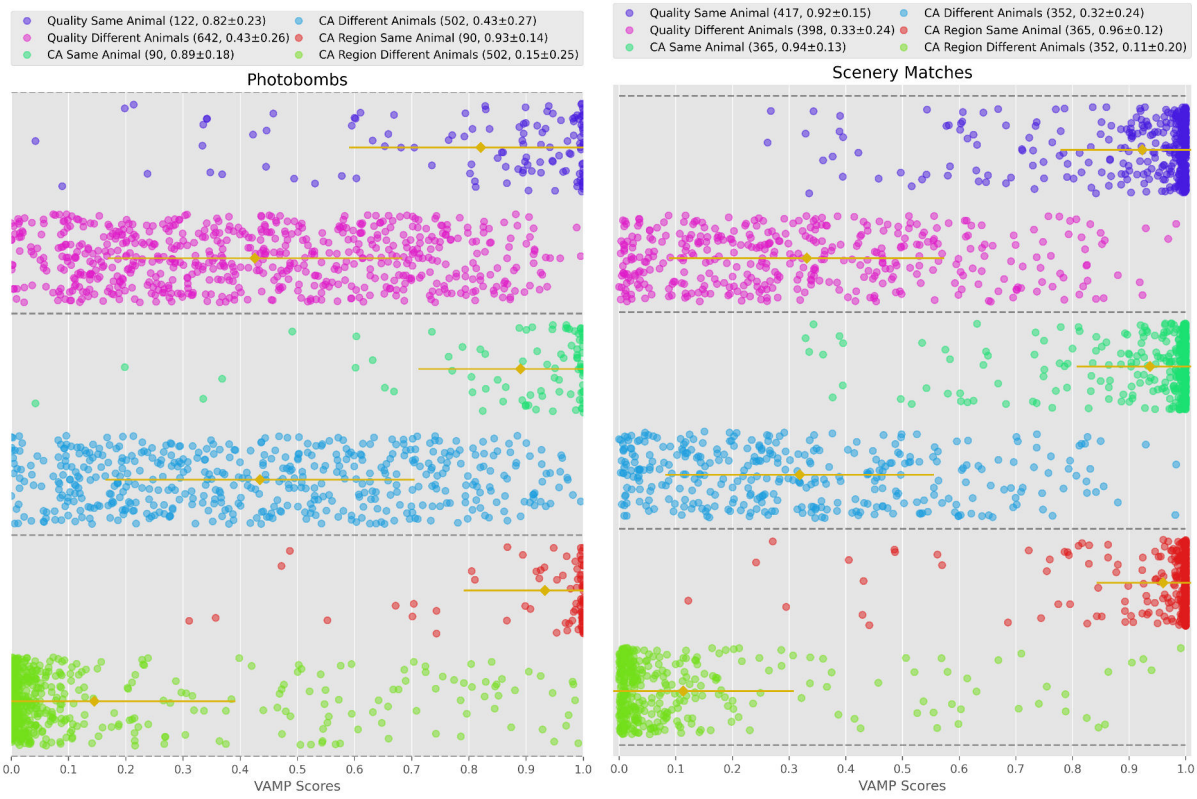


Figure 5.14: A scatter plot of the VAMP scores and their separability for three datasets. The benefit of using Census Annotation Regions over traditional annotations is that it limits the area that is matching to only the identifying information on the body of the animal, decreasing the chance of a photobomb (left) and scenery match (right). The separability of photobomb match scores dramatically improves when Census Annotation Regions (bottom section) are used, with positive pairs scoring 93% and negative pairs scoring 15% on average. Scenery matches also see a dramatic improvement, with positive CA-R pairs scoring 96% and negative pairs scoring 11%.

and cannot be matched. The CA VAMP model classifies the photobomb CA pair as 89% positive (`match`) whereas the CA Region VAMP model run on the CA Region pair predicts 94% negative (`nomatch`). This example shows that CA alone is insufficient to catch all photobomb errors and provides an example of why a perfect detection – that captures the tail in its full detail – is the wrong decision for automating a photographic census with visual ID.

5.6.1.1 Mother-Foals

The human decisions collected by the GZCD did not require explicit labels for mother-foal photobombs when encountered. When the LCA algorithm was used to identify ground-truth ID

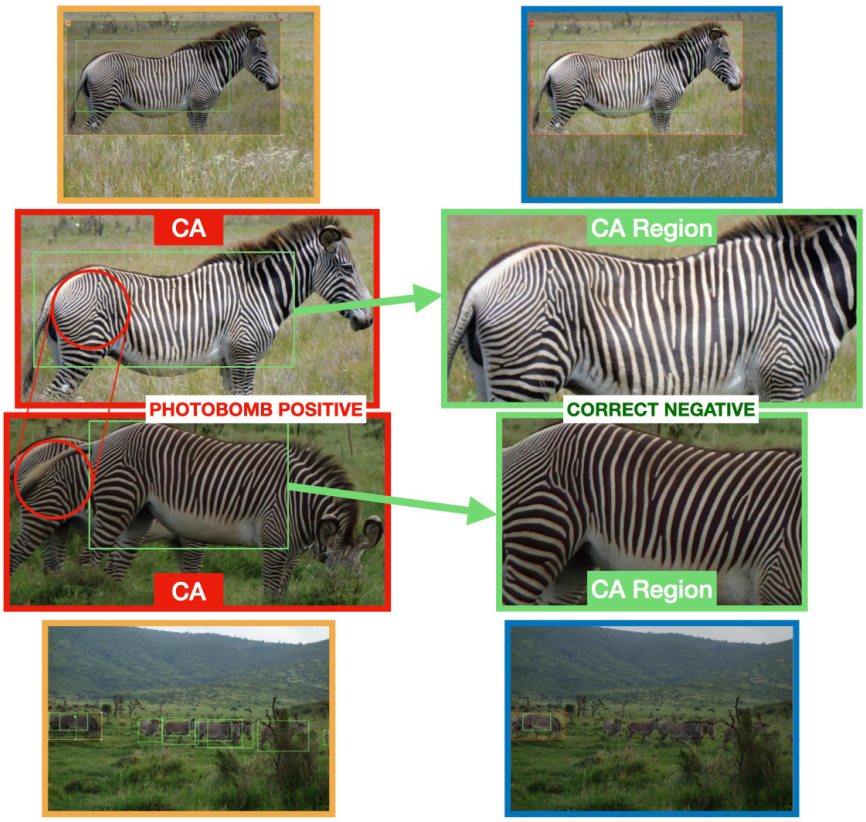


Figure 5.15: An example of a photobomb match that is mitigated by using Census Annotation Regions. The original images (yellow border) with all of the annotations and the highlighted annotation (blue border) matched visually. The annotations are both Census Annotations (red) but still result in a photobomb (red). The matched area (red circles) leads to a likely false positive by an automated classifier. Using the Census Annotation Region (green) shows that the background annotation is removed from visual matching.

errors, however, four animal IDs were found that contained annotations for both a mother and its foal. This type of ID error indicates that – at some point during ID curation – the annotations for the mother and foal were the subject of incidental matching, and their IDs were incorrectly merged. An example of one of these four ID errors is shown in Figure 5.16. We can see that the Census Annotation Regions for the foal and mother overlap significantly. All annotations for the incorrect animal ID are reviewed by hand to separate which annotations show the foal and which show the mother, generating two new animal IDs as the fix.

To fundamentally mitigate mother-foals photobombs, more advanced and nuanced approaches are needed. While these techniques are not evaluated here, using an annotation age classifier or an instance segmentation algorithm will help separate highly overlapping sightings like these. However,

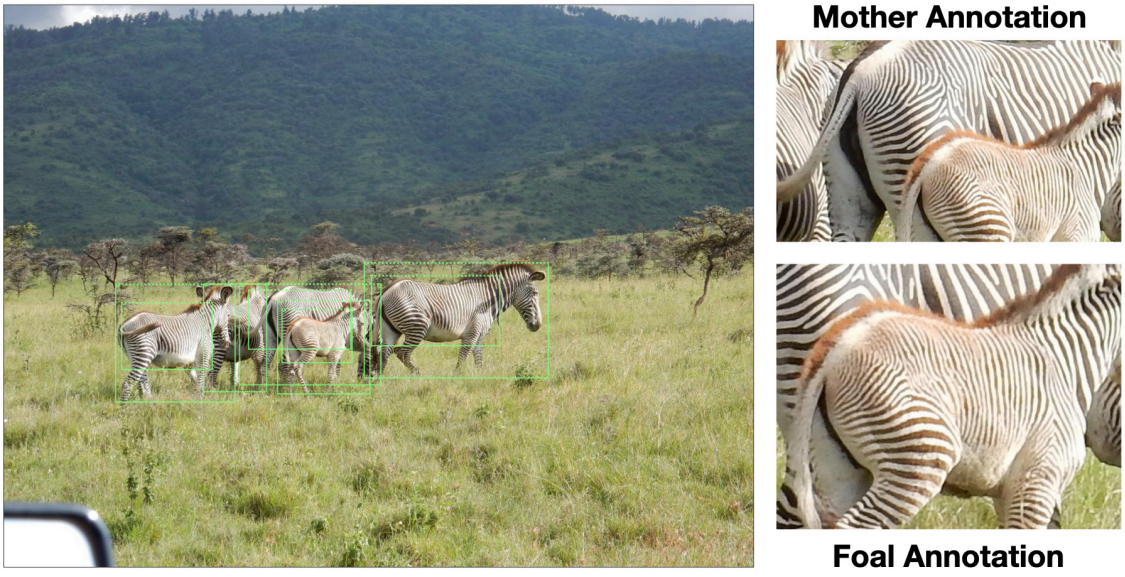


Figure 5.16: An example mother-foal photobomb that was found during the ID curation of the GZCD. The Census Annotation Regions for a foal and mother overlap significantly and are subsequently incorrectly matched. All of the annotations for the merged animal ID are reviewed to separate which annotations show the foal or the mother.

these methods are not evaluated because the relative frequency of mother-foal photobombs seems to be low (four compared to many dozen identified photobombs during the ID curation of the GZCD) when CA-Rs are used for censusing. In addition, more complex segmentation algorithms will place a high burden on annotating ground-truth pixel data. In contrast, the impact of scenery matches can be evaluated without more specialized methods and will be discussed next.

5.6.2 Scenery Matches

The GZCD data contains 1,007 scenery match pairs, with 522 pairs containing the same ground-truth ID (positive pairs) and 458 showing different animals (negative pairs). Since scenery matches are much more random and can happen irrespective of if the match is the same animal or not, a roughly even split is expected (52% actual). Scenery matches happen most often when the photographer(s) take pictures within seconds of each other and without moving the camera's field of view, allowing for the background scene to repeat across images. The annotations were filtered based on the Quality Baseline, using 815 scenery match examples and 717 annotations for the CA algorithms.

As we can see in Figure 5.14 (right), the 417 "same animal" examples for the Quality Baseline

are clustered and classified reasonably well with an average prediction of 92±15%. Using Census Annotations improves the classification accuracy slightly to 94±13% and even better with Census Annotation Regions to 96±12%. Using the Census Annotations improves positive matches when scenery matches are present, but this case is not where most of the errors originate. The "different animals" matches for the Quality Baseline are much more spread out with an average VAMP score of 33±24% and matches the Census Annotation scores at 32±24%. As with photobombs, this is not a surprising failure of the Census Annotation classifier by itself. Moreover, the number of CAs does not decrease significantly from the Quality Baseline to the CA set, considering the relative matchability of the background textures is separate and independent from the comparability of the foreground animal. However, Census Annotation Regions significantly improve negative match separability with an average VAMP score of 11±20%. By repeating the exercise from the photobomb discussion and using a decision threshold of 50%, the Quality Baseline achieves an accuracy of 86.5%, 87.9% for Census Annotation accuracy, and 94.7% for Census Annotation Regions. Picking an optimal operating threshold that maximizes all accuracy for each annotation set yields: Quality Baseline (OP 83%, accuracy 91.2), CA (OP 72%, accuracy 92.9%), CA Region (OP 71%, accuracy 96.5%). These results indicate that scenery matches cases are 1) *not* significantly reduced in quantity by CA filtering but 2) are significantly easier to classify correctly with Census Annotation Regions.

Lastly, we wish to put all of these preceding Census Annotation results together to simulate its impact on human decision-making. The use of CA and CA-Rs has been shown to improve the speed of human verification, the separability of automated decisions, and reduce the impact of incidental matching. Thus, the next and final section shows how using CA and CA-Rs with photographic censusing results in a similar population estimate while requiring a lot less work from humans.

5.7 Population Estimate Simulations

The primary motivation of Census Annotations and Census Annotation Regions is to improve the automation of a photographic census. The discussion has demonstrated that 1) humans spend less time reviewing pairs of CAs and CA-Rs, 2) the scores of the VAMP automated verifier are more separable for CAs and CA-Rs than non-CAs, and 3) the frequency of incidental matching is significantly reduced with smaller CA-R bounding boxes. We still need to determine if CA and CA-R significantly reduce the number of required human decisions and if this reduction in effort causes any loss of accuracy in the final population estimate. Human effort is an important metric to

Table 5.2: The number of annotations, names, singletons for three ID evaluation sets and their GGR-16 and GGR-18 Lincoln-Petersen indices. The "Quality Baseline" set is a traditional filter on species, viewpoint, and quality. In contrast, the Census Annotation (CA) and Census Annotation Region (CA-R) annotation sets (identical numbers below) rely on using a more focused definition of comparability.

Set Name	Annots.	Names	Singletons	GGR-16 L-P	GGR-18 L-P
CA-R	4,142	468	51	366±27	373±29
CA	4,142	468	51	366±27	373±29
Quality	4,269	487	62	360±27	399±29

track because it directly measures how feasible photographic censusing will be for real-world use and is a consistent method for comparing algorithm configurations. Furthermore, the discussion in Chapter 4 gave examples of why automated machine learning algorithms introduce errors and how a more comprehensive review can mitigate them – directly associating an increase in accuracy with an increase in work. Therefore, we can expect that different algorithms, which may be susceptible to different failure methods, can meaningfully influence the accuracy of a population estimate by changing the total amount of human interaction needed. This section simulates various photographic censusing configurations and analyzes their respective accuracy as a function of automation. The simulations use the ground-truth ID data from the GZCD and demonstrate that CA and CA-R reduce the need for human involvement while also producing consistent population estimates.

Before we begin, let us review how the GZCD was constructed (see Section 5.1 for a full description) because it will be the source of ID data for the following simulations. The ID database was built from two sets of annotations: 1) annotations that passed a "species, viewpoint, quality" filter and 2) annotations that were above a specific CA classification score (0.001). The purpose of combining these two sets was that it provided an extensive collection of easy and hard annotation matches and a real-world example of animal sightings. The first collection used the ground-truth labels for species (Grévy's zebra), viewpoint (any viewpoint that contained *right*, and quality (*ok* or better) as its filter; these annotations will be referred to as the "Quality Baseline" set for simulation. This set of annotations is critical to distinguish as a comparative baseline because it is representative of the annotation filtering methods used in prior photographic censusing studies (see [2], [356], and [357]). The annotations that scored above a Census Annotation (CA) threshold of 31% (the recommended value for Grévy's zebra) were selected as the second evaluation set for simulation. Furthermore, each of these CAs were hand-annotated with ground-truth Census Annotation Regions

(CA-R), which formed the simulation's third evaluation set of annotations. In summary, the "Quality Baseline" set has 4,269 annotations for 487 IDs (62 singletons), and the CA and CA-R sets have 4,142 annotations for 468 IDs (51 singletons). Table 5.2 shows a side-by-side comparison of the three sets that will be used for all simulations below. The reason the number of ground-truth IDs between these sets is different is quite simple: their respective IDs are generated from slightly different sets and quantities of annotations. For example, half of the difference in the number of IDs is due to singletons, where there are 11 additional singletons in the "Quality Baseline" set compared to the CA and CA-R sets.

Since photographic censusing relies on sampling, the exact number of ground-truth IDs is not very meaningful for a direct comparison. A better way to contrast different simulated animal ID curation results is to examine the differences in their respective Lincoln-Petersen population estimates. Recall that the GZCD was constructed with images taken in Meru County, Kenya during the Great Grévy's Rally (GGR) photographic censusing events in 2016 (GGR-16) and 2018 (GGR-18). The GZCD is a useful dataset for simulating ID curation approaches because it offers two ground-truth population estimates: one for 2016 and a second independent one for 2018. Reviewing the values in Table 5.2, the Lincoln-Petersen index for the "Quality Baseline" is 360$\pm$27 in 2016 and 399$\pm$29 for 2018. The population estimate based on only Census Annotations was 366$\pm$27 for GGR 2016 and 373$\pm$29 for GGR 2018. Finally, simulations with Census Annotation Regions estimate 366$\pm$27 zebra were within Meru County in 2016 and 373$\pm$29 animals in 2018. These values will function as the "targets" for each of the following simulations, conditioned on their input annotation set.

5.7.1 Which Annotations to Select

Six photographic census events were simulated to measure the impact of how annotations are selected: the Graph ID and LCA algorithms were simulated on the three annotation sets, respectively. All of the simulations in this sub-section used HotSpotter for the ranking algorithm and VAMP as the automated verifier. Each decision management algorithm was provided with the same underlying ranked list from HotSpotter. Hotspotter was configured to use $K = 5$, $K_{\text{norm}} = 5$, and had spatial verification turned on (as recommended by [13] for Grévy's zebra). These values control the number of Approximate Nearest Neighbor matches returned for each SIFT keypoint and determine how to normalize their respective match scores. Furthermore, the ranked list was configured to return the 10 highest-scoring annotations ($n_{\text{top}} = 10$) for each sighting. The "Quality

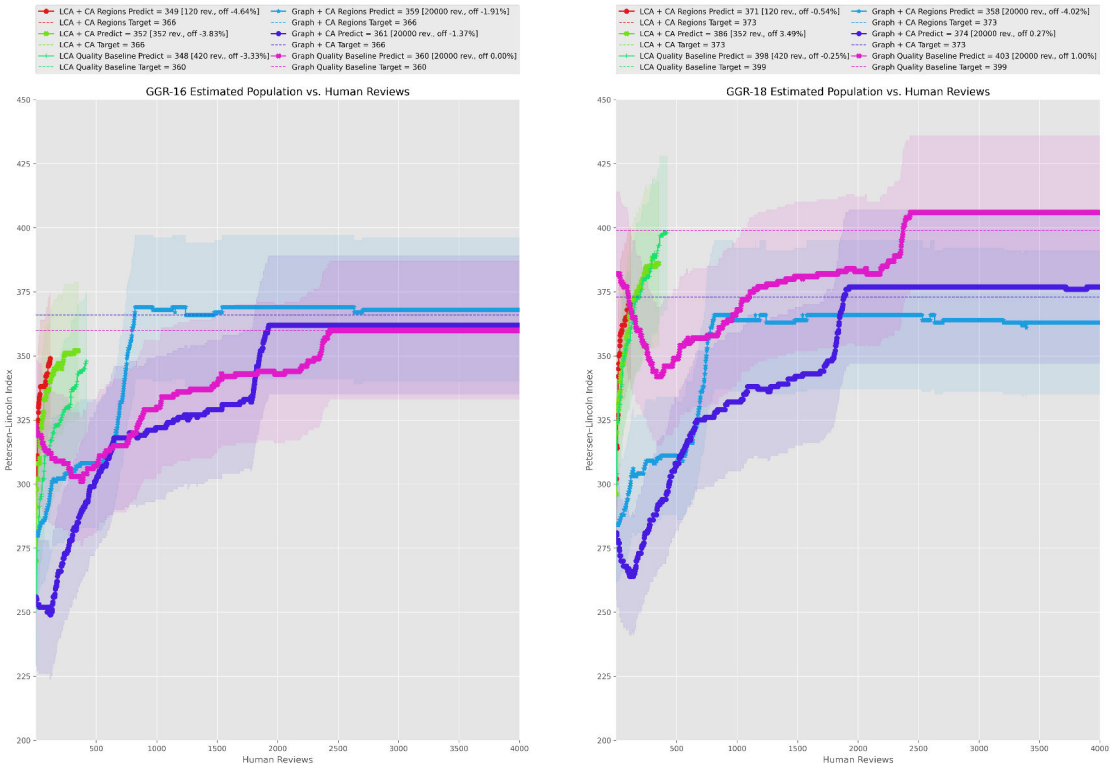


Figure 5.17: The simulated population estimates over 4,000 human decisions. Simulated Lincoln-Petersen population estimates are shown on the y-axis against the number of human decisions were requested on the x-axis. The estimates for GGR-16 (left) and GGR-18 (right) are shown for three sets and with two separate graph curation algorithms.

Baseline" set used 27,075 pairs from HotSpotter's LNBNN ranking algorithm, and the CA and CA-R sets were provided with 25,831 ranked pairs. When LCA received these pairs, it re-scored them with its internal weighting function (i.e., the "weighter") and predicted 17,862 positive edges for the "Quality Baseline" set, 17,205 for the CAs, and 17,749 for CA Regions. Each of the LCA weighters was initialized using a VAMP [13] model that was trained for each set independently and used the scores for 1,000 randomly sampled *"same animal"* and 1,000 *"different animals"* ground-truth decision pairs. The LCA simulations were allowed to run to convergence (configured with a minimum delta convergence multiplier of 1.5 and minimum delta stability ratio of 4), but the Graph ID algorithm (configured for a positive redundancy of 2 and a negative redundancy of 1) was stopped prematurely after 20,000 human decisions were requested.

When a human decision was needed during the simulation, the decision was obtained by querying a *perfect human oracle* (i.e., 100% accuracy) that inferred the correct answer from the

ground-truth IDs. The oracle made perfect decisions because the goal here is to compare methods of annotation selection without needing to worry about randomness introduced by a fallible human reviewer; later experiments will examine the impact of human error in the ID curation process. Both decision management algorithms started with fresh initialization states and were not given information about previous pair decisions used to create the ID dataset. Each algorithm, however, was free to request as many automated and human pair decisions as it wanted during its ID curation. Each time a human decision was requested, the current state of the algorithm's population estimate was recorded. A log included the current number of IDs in the simulation's database, what the GGR-16 and GGR-18 population estimates were at that moment, and the number of total automated reviews requested since the start of that simulation run. Figure 5.17 shows the GGR-16 (left) and GGR-18 (right) population estimates for the GZCD as a function of human decisions. Note that a fluctuating and an indeterminate number of automated reviews may have been requested between subsequent human reviews, which is not plotted.

The format of the plot in Figure 5.17 is something this discussion will rely on extensively, so it is essential to understand what is being displayed and compared. Fundamentally, the figure shows the population estimate (y-axis) as calculated in real-time after every human decision (x-axis) that was requested by a given curation configuration. The pink dashed lines on both plots represent the "Quality Baseline" annotation set and its associated population estimate. The dashed blue lines represent the population estimates calculated using the CA and CA-R annotation sets. Thus, the dashed lines represent the target that each simulation is trying to approximate. For example, the GGR-18 simulations on CA annotations should ideally produce 373 individuals, as indicated by the dark blue dashed line.

Something we must consider is how consistent the ground-truth population estimates are for a given input. The GGR-16 Lincoln-Petersen index using any of the three annotation sets is approximately 360 or 366, a spread of less than 2%. The GGR-18 estimates are farther apart but are still relatively close (off by 26). The ground-truth estimates with the CA and CA-R annotations are 373 IDs, while the "Quality Baseline" estimated 399 animal IDs (7% spread). While the population's actual number is unknown for both years, the GZCD has 350 ground-truth IDs for 2016. If we were to consider the actual value of 366 (i.e., halfway between the two estimates), it implies that approximately 96% of the surveyed animal population was seen. A high percentage of ID coverage indicates that the photographers thoroughly saturated the survey area and produced a highly representative sampling of its population. The effect is that the confidence intervals (95%)

for the ground-truth population estimates are relatively compact (less than ± 30 IDs). Likewise, the ID database has 367 ground-truth IDs for 2018; if the actual size of IDs was 386, then the photographers sampled around 95% of the population. Furthermore, these coverage percentages are very high, and the number of resightings between day 1 and day 2 is also very high (approximately 220 for GGR-16 and 195 for GGR-18). The takeaway is that while the actual number of animals is unknown, all simulations' confidence intervals overlap in their respective target population estimates and confidence intervals.

Focusing on the Graph ID performance curves for GGR-16 and GGR-18, we can see the excellent filtering effect of CA (dark blue line) vs. the "Quality Baseline" (magenta). For the GGR-16 results, the baseline algorithm asymptotically approaches the correct number at around 2,500 human reviews, while the CA curve shows a 20% savings in the number of human reviews. The quality baseline estimate is precisely correct and predicts 360 individuals, while the CA prediction of 361 under-estimates its target of 366 by 1.4%. Likewise, the GGR-18 results show a similar savings of roughly 20%, but the accuracy improves slightly (1% error with quality to 0.3% error for CA) when comparing against their respective targets. This result indicates that using Census Annotation alone as a high-level classifier can speed up the convergence of the Graph ID algorithm while not sacrificing any meaningful accuracy in the estimate. Comparing these results to using the Graph ID algorithm (and corresponding VAMP model) on Census Annotation Regions shows an even more drastic reduction in the number of human reviews. The algorithm converges at around 700 human decisions (a reduction of over 70% compared to the Quality baseline) and ends up being incorrect in its estimate by only 1.9%. The story for GGR-18 is similar as it also concludes at roughly the same number of reviews but does under-estimate the number of animals by 4.0% (15 names). It is reasonable to expect that being off by less than 5% is acceptable for this application, especially since any bias from machine learning algorithms has not been accounted for yet. It would seem that 5% is well within the margin of error since the confidence interval is larger at around 7%. The Graph ID algorithm better approximates the ground-truth estimate (1% error) for GGR-18 when it uses the quality baseline.

Next is a review of the LCA algorithm simulation results. By design, LCA is focused on delaying human decision-making as much as possible. As a result, it can accurately approximate the population estimate with significantly fewer reviews than Graph ID but requires more up-front processing. This trend is visible in Figure 5.17, as the green and red LCA lines end (where the algorithm converged) at a much lower value on the x-axis compared to the blue and magenta Graph

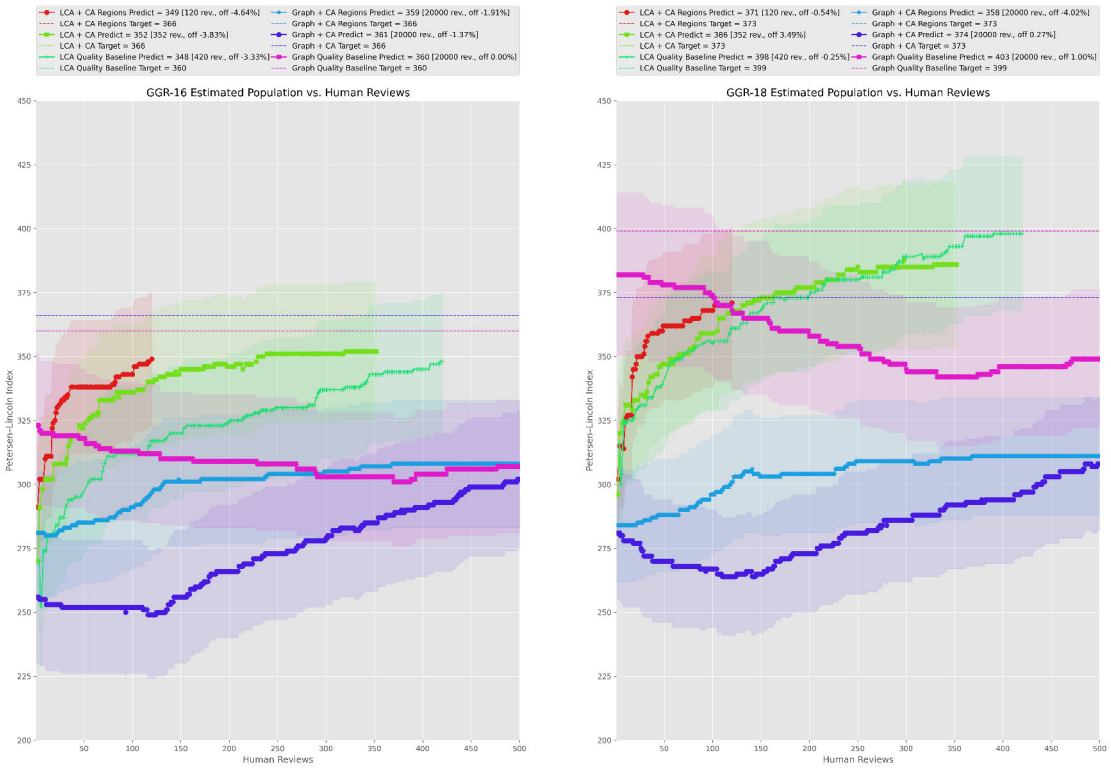


Figure 5.18: The simulated population estimates over 500 human decisions. Simulated Lincoln-Petersen population estimates are shown on the y-axis against the number of human decisions were requested on the x-axis. The estimates for GGR-16 (left) and GGR-18 (right) are shown for three sets and with two separate graph curation algorithms.

ID lines. To better continue the analysis on LCA, a re-scaled copy of Figure 5.17 is provided in Figure 5.18. This second plot shows the same data but has an x-axis covering the range [1, 500] human reviews, which better displays the differences between the three LCA simulations. The LCA algorithm on the Quality Baseline set (green line), as expected, took the most number of human reviews (420) to converge. Because LCA does not explicitly require consistency and makes better use of the automated verifier, it can finish with much less human involvement than the Graph ID algorithm. However, the LCA algorithm does under-estimate the target value by 3.3% (12 names) for GGR-16. Switching to using LCA on Census Annotations gives a final estimate within 3.8% of the ground-truth value but saves 68 human reviews (16% reduction). The Census Annotation estimate also under-shoots the correct value by approximately 3.5%, consistent with the baseline error. This result means that using Census Annotations as a classifier only decreases the relative accuracy by 0.5%. Just as with the Graph ID algorithm, the LCA algorithm drastically improves

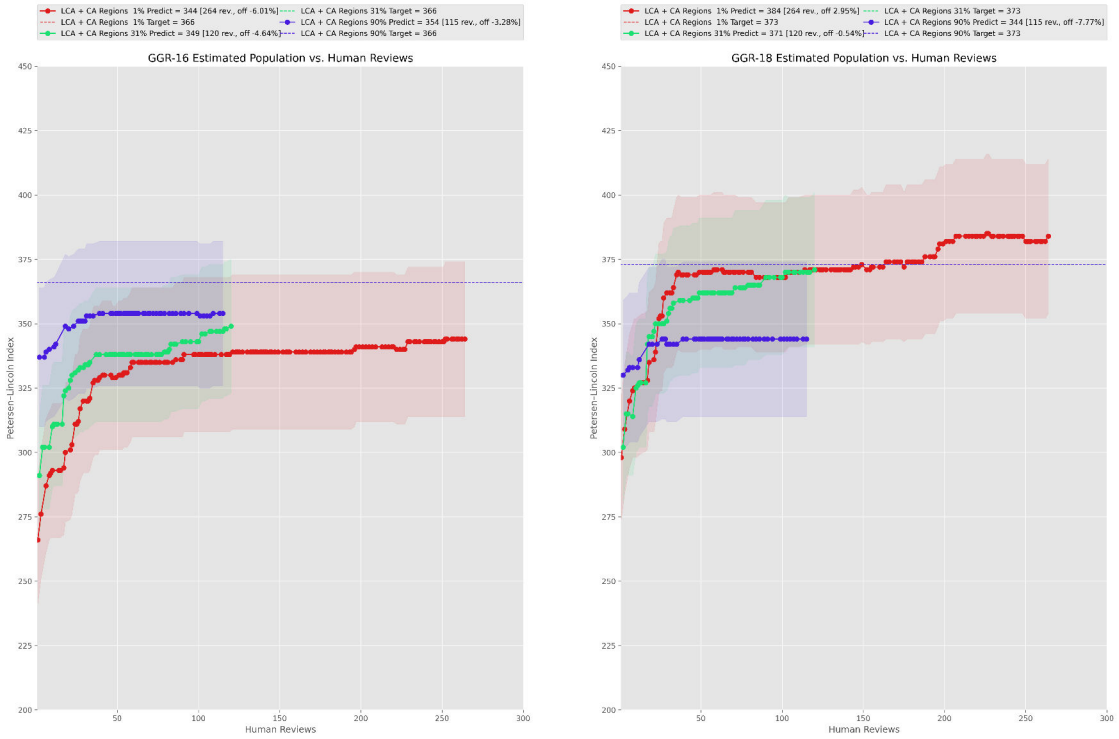


Figure 5.19: The simulated population estimates across different Census Annotation thresholds. Simulated Lincoln-Petersen population estimates are shown on the y-axis against the number of human decisions were requested on the x-axis. The estimates for GGR-16 (left) and GGR-18 (right) are shown for three different sets of Census Annotation Regions, selected with thresholds at 1%, 31% (recommended), and 90%.

automation when Census Annotation Regions are used. The number of required human reviews for the most automated decision management algorithm configuration was only 120 in total and is accurate within 4.6% on GGR-16 data (predicting 349 animals against a ground-truth value of 366). The results for GGR-18 mirror the narrative from the GGR-16 results, with LCA run on Census Annotation Regions predicting the correct answer within 0.5% (under-shooting by two names). The Quality Baseline LCA results were off by only one name but ended up needing nearly four times the number of human reviews.

5.7.1.1 Census Annotation Decision Thresholds

We also can quickly consider what would happen when the Census Annotation thresholds are changed from the recommended 31% to lower and higher values. If the filtering threshold is lowered to 1%, the number of used annotations will increase but for the added risk of incomparable

and incidental matches. As we have established, these errors will result in more overall work and likely bias the estimate higher. On the other hand, a high filtering threshold of 90% will provide very clean and comparable animal sightings to animal ID curation, making it much more likely to miss a relevant animal sighting. Figure 5.19 shows the simulations for these two cases with the LCA algorithm and CA-Rs as the input data. We can see that the 1% population estimates for GGR-16 and GGR-18 (red line) require much more work to converge, and the algorithm achieves a worse result. Likewise, using a very high CA filtering threshold of 90% (blue line) does not increase work compared to the recommended 31% (green line) and under-shoots the target by 7.8% for GGR-18. That being said, the simulated value is still within the confidence interval for those two populations.

In summary, Census Annotation – and more specifically Census Annotation Regions – is a powerful tool for increasing automation through effective filtering. Furthermore, this improvement can be seen with two entirely different decision management algorithms. For example, comparing LCA on Census Annotation Regions to Graph ID on the "Quality Baseline", the latter required over 2,500 reviews to closely approximate the target estimate (and over 10,000 reviews to get within 1% on GGR-18). In contrast, LCA with CA-R required only 120 human decisions to get within 5% of the ground-truth estimate on GGR-16 and 0.5% for GGR-18. We can also see that photographic censusing is somewhat sensitive to how well Census Annotations are filtered. Therefore, we want to remove annotations that cause errors and include as many annotations as possible for accurate estimates. When the CA filtering was at 90%, the estimate decreased. This decrease seems to defy the conclusions of Equation (4.21); the problem is that the threshold is putting pressure on Assumptions 3 and 4 (see Section 4.3.1), and the estimate is therefore not reliable. Next, we compare how much of the overall ID curation work each decision management algorithm is automating with algorithmic verifiers.

5.7.2 Degree of Decision Automation

During each of the six simulations from the previous sub-section, the total number of requested VAMP decisions were recorded after each human decision. Each simulation used HotSpotter as the ranker and VAMP as the verifier but varied on their input annotation sets and the decision management algorithm used. We can contrast the total number of automated verification decisions requested to cross-examine how much of the overall ID curation process was automated. For example, it was shown earlier in this chapter that VAMP automates a higher percentage of decisions (for a fixed FPR) when Census Annotation and Census Annotation Regions are used. Allowing

VAMP to perform more of the overall ID curation work implies that the process also requires less work from a human reviewer.

For the Graph ID simulations, the algorithm either converged or was prematurely stopped after 20,000 human decisions. Even though the algorithm had not officially converged, that does not mean it was not closely approximating the target estimate well before its termination. For each Graph ID simulation, there is a distinct point where the algorithm approximately converges, and it is helpful to track this point for comparison. For the Quality Baseline set, this point is at 2,414 human reviews; for the CA set, it is at 1,923 reviews; and for the CA Regions, it is at 642 reviews. If the Graph ID algorithm were stopped prematurely at each of these junctures, the predicted estimates would have all been within 5% correct. Another way to consider the work done by an automated verifier is to cap its total number of decisions. For example, pretend that the automated verifier was very slow, much slower than a human, or was computationally expensive. In that case, we may want to consider placing a budget on the total number of automated decisions before terminating. For example, a fixed budget of 5,000 automated reviews would mean that each annotation in the three evaluation sets participates in at least two reviews on average. Table 5.3 provides a breakdown for the number of human and VAMP decisions that were needed for each annotation set and decision management algorithm. For the Graph ID algorithm, the table offers the termination points at 20,000 decisions, points where the correct value is approximated, and early-stopping points based on a strictly enforced budget.

The results show that using Census Annotation Regions improves the rate of automation compared to the Quality Baseline. For example, the Graph ID algorithm converged (terminated at 20,000 human reviews) with 50.5% of the total reviews being automatically decided by VAMP. In contrast, Graph ID on CA Regions automated a total of 52.5% of the reviews, a savings of 1,714 reviews. In addition, VAMP is doing more work overall on Census Annotation Regions as there are 5.3 decisions per annotation on average compared to 4.8 for the Quality Baseline. For the Quality Baseline, the VAMP model was able to automate 12,988 reviews, with an automation rate of 84.2% The Census Annotation, while requiring approximately the same number of overall reviews as the baseline set (15,402 vs. 15,110, a difference of less than 2%), required significantly fewer human reviews at 1,923 and an automation rate of 87.3%. Thus, using Census Annotations saved the human reviewers from needing to complete nearly 500 reviews, a 20.3% reduction. Going a step further, Census Annotation Regions require only 5,553 total reviews to approximate the estimate, with only 642 being done by a human. Compared to using Graph ID on the Quality Baseline set, using CA

Table 5.3: The amount of work done by the automated verifier reduces the number of human reviews. For the Graph ID algorithm, a simulation was considered *converged* when the number of requested human reviews exceeded 20,000. The average number of VAMP reviews per annotation in parenthesis is below the number of VAMP Reviews. We can see that using LCA on CA Regions results in the lowest number of human decisions.

Algorithm	Set	Annotations	VAMP Reviews	Human Reviews	Total Reviews	Automation Rate
Graph ID *Converged*	Quality Baseline	4,269	20,374 (4.8)	20,000	40,374	50.5%
Graph ID *Converged*	CA	4,142	19,055 (4.6)	20,000	39,055	48.8%
Graph ID *Converged*	CA Region	4,142	22,088 (5.3)	20,000	42,088	52.5%
Graph ID *Approximated*	Quality Baseline	4,269	12,988 (3.0)	2,414	15,402	84.3%
Graph ID *Approximated*	CA	4,142	13,187 (3.2)	1,923	15,110	87.3%
Graph ID *Approximated*	CA Region	4,142	4,911 (1.2)	642	5,553	88.4%
Graph ID *Budgeted*	Quality Baseline	4,269	5,000 (1.2)	2,255	7,255	68.9%
Graph ID *Budgeted*	CA	4,142	5,000 (1.2)	1,747	6,747	74.1%
Graph ID *Budgeted*	CA Region	4,142	5,000 (1.2)	645	5,645	88.6%
LCA *Converged*	Quality Baseline	4,269	22,552 (5.3)	420	22,972	98.2%
LCA *Converged*	CA	4,142	18,255 (4.4)	352	18,607	98.1%
LCA *Converged*	CA Region	4,142	13,307 (3.2)	120	13,427	99.1%
LCA *Budgeted*	Quality Baseline	4,269	5,000 (1.2)	121	5,121	97.6%
LCA *Budgeted*	CA	4,142	5,000 (1.2)	92	5,092	98.2%
LCA *Budgeted*	CA Region	4,142	5,000 (1.2)	63	5,063	98.8%

Region with the same algorithm results in a 73.4% reduction in human work.

Furthermore, the LCA algorithm with the Quality Baseline set is highly automated. In total, it makes 22,972 decisions and asks for only 420 of those from a human (automation rate of 98.2%). The LCA algorithm is designed to try alternative clustering of the current ID graph, which calls for an automated or human decision. The algorithm's degree of confidence in a particular clustering can be partially seen in how many reviews are requested before the algorithm converges. This behavior means that the algorithm will converge faster if the decisions are coherent, the annotations are discriminative, and the edge weights are stable. For example, the LCA algorithm converges faster with Census Annotations with a total number of reviews of 18,607 (automation rate of 98.1%); the algorithm can converge with 4.4 decisions per annotation compared to 5.3 with the quality baseline, indicating that the former was more discriminate and required fewer alternative clusterings. The CA Regions shows a dramatic reduction in the number of total reviews at 13,427 while only requiring 120 from a human (99.1% automated). This result represents a 71.4% reduction in human work and an even lighter workload at 3.2 decisions per annotation.

Considering a budget restriction with the Quality Baseline set, the Graph ID algorithm only automated 68.9% of the reviews. In contrast, using Census Annotation automates 74.1%, and using Census Annotation Regions achieves a rate of 88.6%. A similar – albeit less dramatic – improvement in automation occurs using the LCA algorithm; using CA Regions and a budget cuts the number of human decisions in half (to 63) compared to the Quality Baseline of 121. In summary, these results indicate that Census Annotation Regions improve automation for both the Graph ID and LCA decision management algorithms. Furthermore, LCA can make much more efficient use of the automated verifier than Graph ID.

5.7.3 Comparison of Automated Ranker & Verifier

We next consider the impact that different ranking and verification algorithms have on photographic censusing. As discussed in Chapters 2 and 4, the HotSpotter algorithm was developed alongside the detection pipeline presented here, the VAMP verifier, and the Graph ID decision management algorithm for the problem of estimating Grévy's zebra populations. Therefore, it is apparent and unsurprising that most of the photographic censusing analysis presented in this dissertation relies on it heavily. This preference is due to 1) its ability to match sightings without any species-specific training, 2) it does not rely on extensive ground-truth ID data to bootstrap, and 3) it is fast to produce accurate results. For similar reasons, the VAMP verifier is used thus

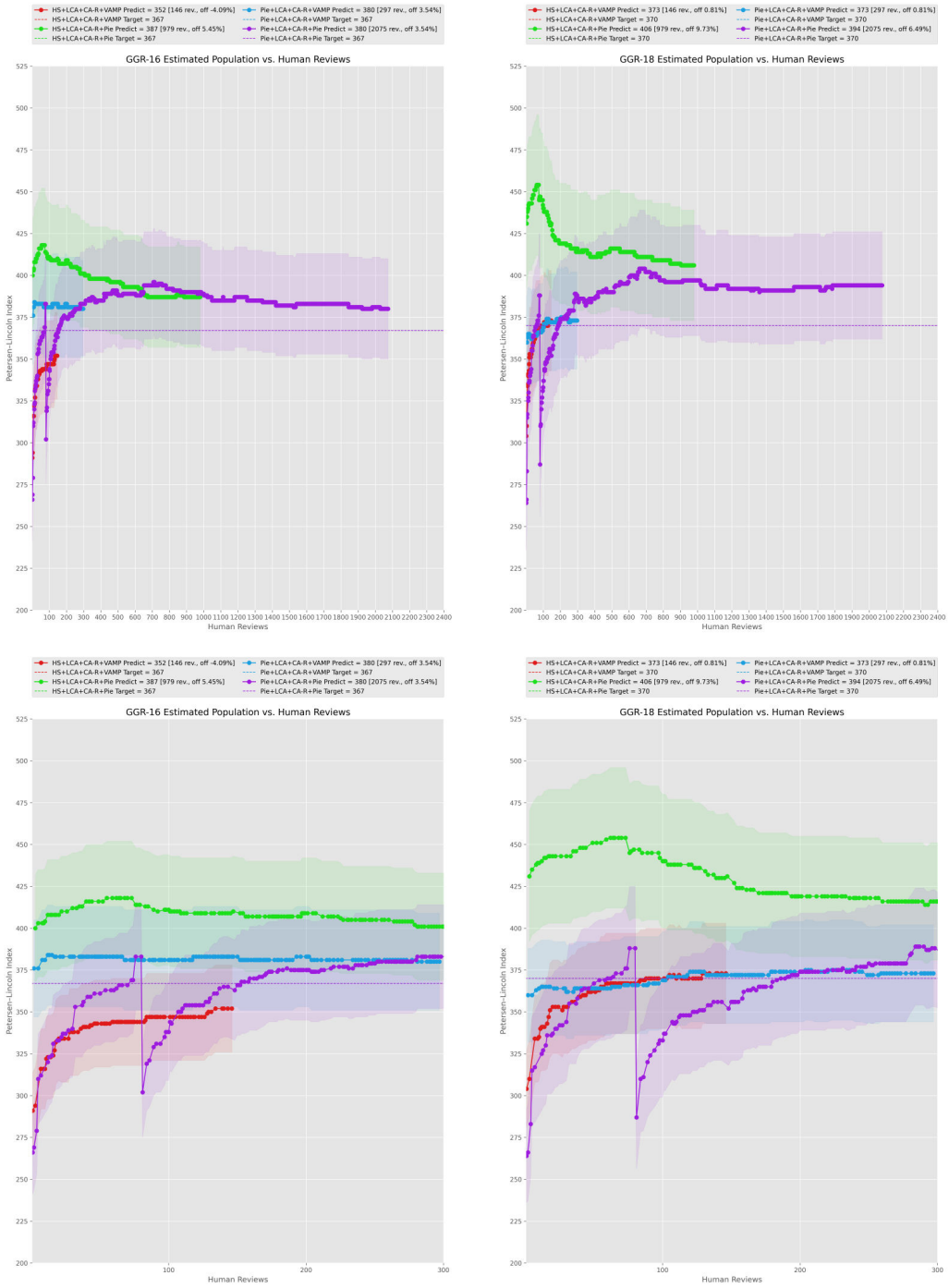


Figure 5.20: The simulated population estimates with different ranking and verification algorithms. Simulated Lincoln-Petersen population estimates are shown on the y-axis against the number of human decisions were requested on the x-axis. The estimates for GGR-16 (left) and GGR-18 (right) are shown for different ranking and verification algorithms.

far to automate human decision-making for Grévy's zebra. Even though both approaches are the presumed defaults, they are not the only options available for the automated ranking and verification tasks. Specifically, the PIE [263] algorithm can be trained to perform both tasks (ranking and verification) as a triplet-loss neural network.

The PIE algorithm is helpful to photographic censusing for many reasons: 1) it has a superior feature extraction process compared to HotSpotter (which is based on the outdated SIFT algorithm), 2) it is exceedingly fast (and supports GPU acceleration during training and inference), and 3) it is highly accurate for known, consistent poses of an animal (e.g., a right-facing Census Annotation Region). Furthermore, the PIE algorithm produces a single, fixed-length feature vector per annotation, which can be quickly computed, cached, and compared in L2 space for fast matching (e.g., clustering with approximate nearest neighbors) and verification (e.g., a distance threshold). The downside is that PIE requires an existing ID database to train and cannot be bootstrapped without prior IDs. After building a relatively large ID dataset like GZCD with HotSpotter, however, PIE can be trained to perform accurate matching and verification. The model can also be used to identify ground-truth ID errors that HotSpotter was unable to recognize, improving the reliability of the ID database and providing even better data for re-training the PIE algorithm.

With CA-R for input annotations, HotSpotter for ranking, VAMP for verifying, PIE for ranking and verifying, LCA for curation, simulated humans for verifying, and Lincoln-Petersen for estimating the population, we have all of the tools at our disposal for robust photographic censusing. Figure 5.20 shows a comparison of four simulations:

1. **Red Line** - Input: CA-R, Ranker: HotSpotter, Verifier: VAMP, Curation: LCA
2. **Green Line** - Input: CA-R, Ranker: HotSpotter, Verifier: PIE, Curation: LCA
3. **Blue Line** - Input: CA-R, Ranker: PIE, Verifier: VAMP, Curation: LCA
4. **Purple Line** - Input: CA-R, Ranker: PIE, Verifier: PIE, Curation: LCA

The figure shows GGR-16 (left) and GGR-18 (right) results for two x-axis scales. The top row shows the same result for a maximum of 2,400 human decisions, and the bottom row uses a maximum of 300 decisions.

All four simulations end with a population estimate within 5.5% for GGR-16 and 10% for GGR-18. The clearest outliers are for the simulations where PIE operates as the verifier, consistently over-shooting the target. Upon inspection, PIE is performing so poorly because it is sensitive to

changes in pose and comparable quality. The majority of the extra IDs that PIE suggests is where it takes a sizeable single ID and splits it into two IDs, with one ID containing only 1-2 annotations and the second ID containing all other annotations for that ground-truth ID. This result suggests that PIE can be helpful to identify outlier Census Annotation Regions that are on the border of being comparable. Furthermore, for GGR-16 and GGR-18, the PIE ranking algorithm with VAMP as the verifier was the most accurate configuration (3.5% error and 0.8%, respectively). Compared to HotSpotter with VAMP, however, that best result came at the cost of double the number of human reviews.

In summary, the HotSpotter and PIE algorithms work well as ranking algorithms for Grévy's zebra. However, using PIE for animal ID curation inflates the overall estimate because pose variations are challenging for that algorithm, and careful attention should be paid to animal IDs with only a few annotations. We will conclude this chapter by examining the impact human decision errors have on animal ID curation.

5.7.4 Effect of Human Verification Accuracy

All of the simulations above have used a perfect human oracle whenever a manual decision is needed. This oracle was configured to have a guaranteed accuracy of 100% so that any effects from human fallibility are removed, and the different algorithms can be compared in a more standardized environment. The expectation that the human is always perfect is unrealistic, however, even for Census Annotation Regions. Looking back to the user study in Section 5.4, we can recall that an expert human reviewer's accuracy is around 98% on average for general pairs of annotations, and novice reviewers are approximately 94% accurate. The lowest accuracy measured for a human reviewer during that study was 91.7%, so a minimum expectation of 90% seems realistic as a test condition. To compare the impact of poor human decision-making, we can simulate the human oracle with varying levels of random error for a fixed animal ID configuration. The simulations in this sub-section were completed with CA Regions as the input annotation set, HotSpotter as the ranker, VAMP as the verifier, and LCA as the decision management algorithm to keep the comparisons simple.

Figure 5.21 shows the simulated results with a human error rate ranging from 50% to 100%. All previous simulation "baseline" results for this configuration are shown as a red line and converge after 120 human decisions. As expected, when the accuracy of the human verifier drops, the total number of reviews increases. Encouragingly, however, the number of human decisions only

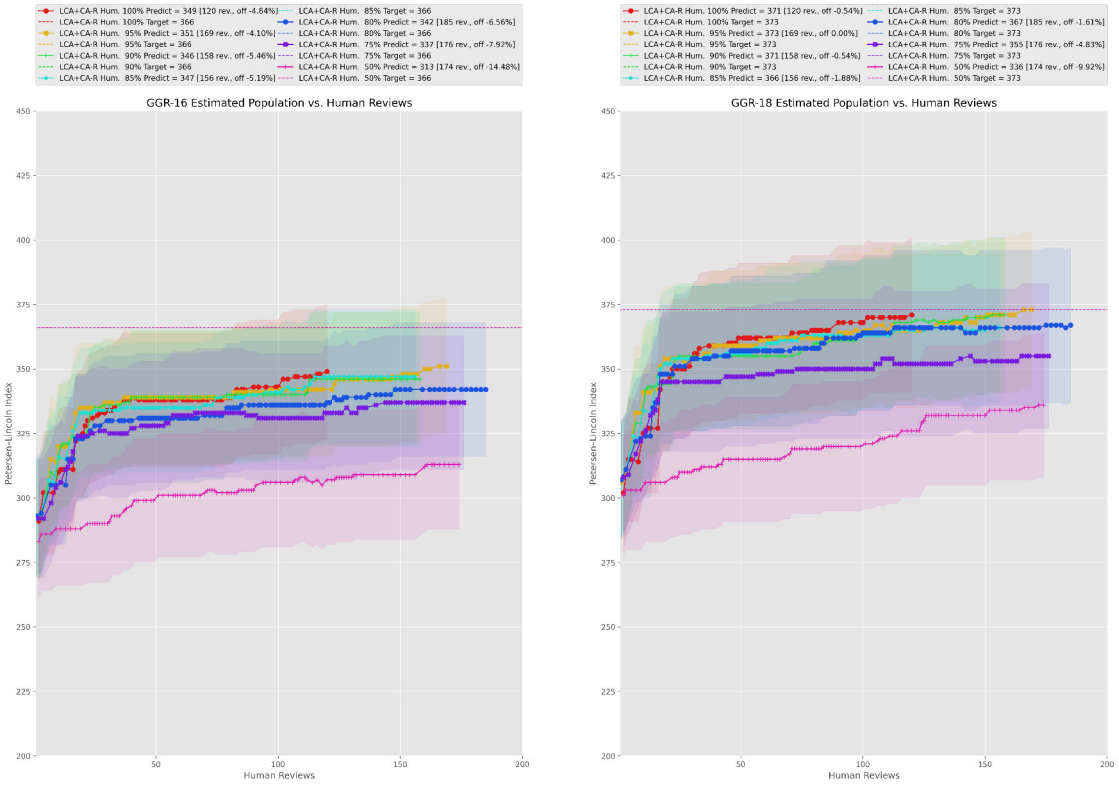


Figure 5.21: The simulated population estimates across different human accuracies. Simulated Lincoln-Petersen population estimates are shown on the y-axis against the number of human decisions were requested on the x-axis. The estimates for GGR-16 (left) and GGR-18 (right) are shown for different simulated levels of human decision accuracy, ranging from 50% to 100%.

increases by about one-third. This effect is seen down to a human accuracy of 85% and achieves the same result as the 100% simulation. This result is hopeful since it suggests that photographic censusing with LCA is fairly fault-tolerant and can accept human error with a healthy margin to spare for an individual user's ability. At 80% human accuracy, we start to notice a decrease in overall accuracy by 2-4% and a significant drop-off in performance at 75%. Interestingly, the LCA algorithm still converges when the human verifier is 50% accurate (a coin flip). The problem is that the GGR-16 population estimate is 14.5% under-counting, and the GGR-18 estimate is incorrect by -10%. An accurate human verifier is therefore crucial to the reliability and automation of photographic censusing. These effects, while not tested, will only be exacerbated with the CA and "Quality Baseline" sets and will be much more devastating to reducing the overall workloads with the Graph ID algorithm. As the best-case scenario, the comparison provided here demonstrates that a minimum human accuracy of 90% is an acceptable target for practical use. Similar to the loss

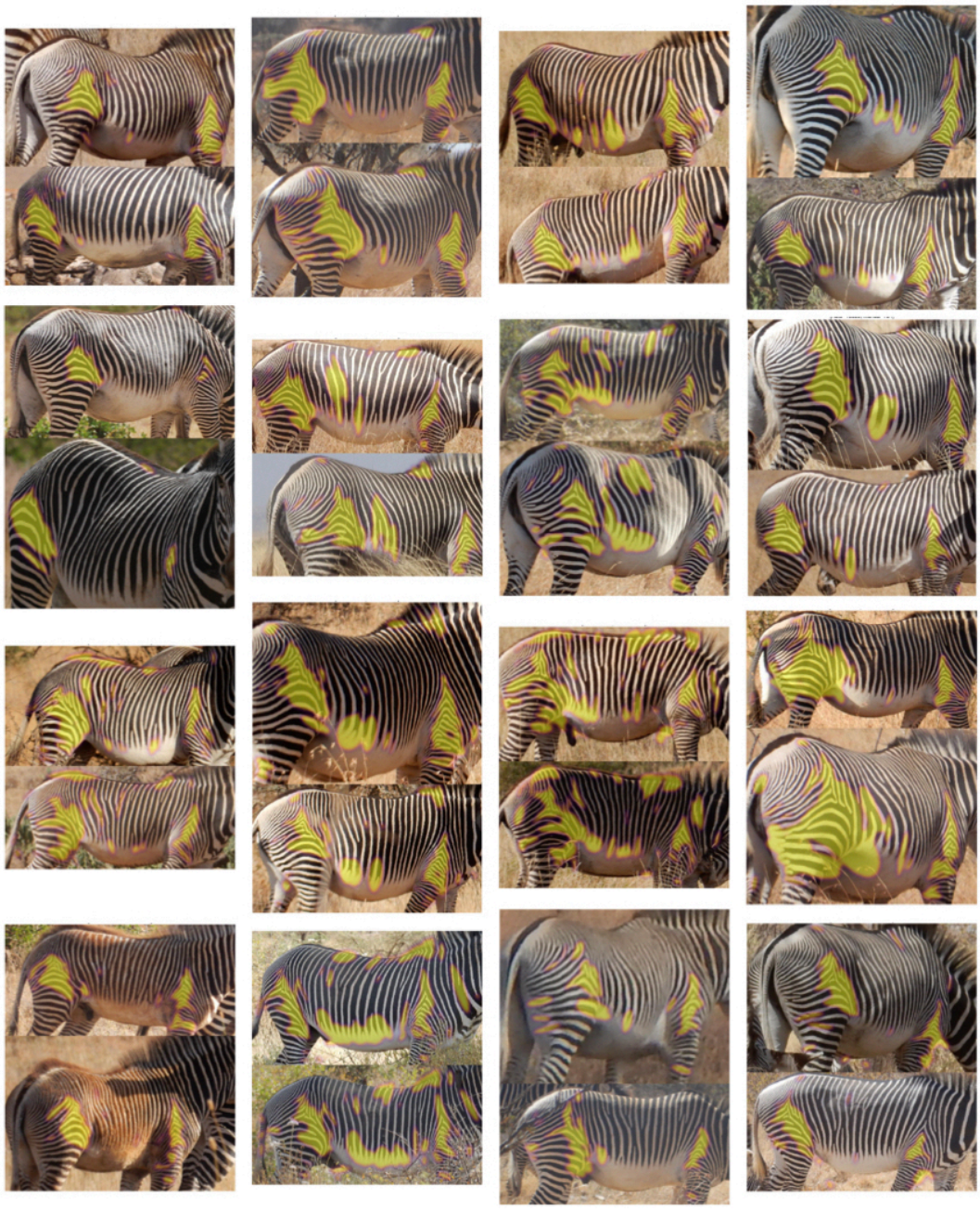


Figure 5.22: Example images of HotSpotter matched regions (in yellow) for Grévy's Zebra. The HotSpotter algorithm automatically finds corresponding texture patterns between two images and ranks likely matches. The regions that tend to match strongly for Grévy's zebra are the hip and shoulder areas, which are highlighted uniformly for all examples. The concepts of Census Annotation and Census Annotation Regions (also shown here) are designed to focus a photographic census on the most likely matching areas while also removing distracting background textures from plants and animals.

of accuracy when the CA classifier was run at 90%, if the accuracy of the human reviewers drops too low, then Assumption 3 (see Section 4.3.1) begins to break as the inter-encounter recall is no longer perfect.

5.8 Summary

Census Annotations and Census Annotation Regions are critical to automated photographic census because they 1) speed up human verification of match pairs and reduce the number of manual decision errors, 2) better separate the positive and negative scores predicted by algorithmic verifiers, 3) reduce the number of photobombs and scenery matches, and 4) drastically reduce the amount of human interaction needed during animal ID curation. Census Annotation Regions are powerful because they force a photographic census to consider only the most critical information for matching, as seen in Figure 5.22.

In summary, human verifiers can make 40% more decisions in the same time frame when comparing Grévy's zebra CA-Rs and make 70% fewer mistakes. However, human verifiers are also known to make mistakes up to an error rate of 10%. A human accuracy rate of 90% has been shown with simulations to only increase work by approximately 30% and end with a consistent result (compared to a perfect verifier). The VAMP verification algorithm can automatically decide – given a maximum false-positive error rate – 10% more pair decisions when CAs are used and 15% more with CA-Rs, compared to using the standard annotations produced by the detector. Furthermore, CA-Rs dramatically improve the automated score separation for known cases of incidental matching, increasing the distance between the average positive and negative scores by 77% for photobombs and 84% for scenery matches.

Finally, simulations with Census Annotations and Census Annotation Regions demonstrate that using those annotations results in consistent population estimates compared to a known baseline. For example, a photographic census with CA-Rs uses 70% fewer human decisions than the quality baseline and only increases the population estimate error by 0.5% for GGR-16 and 0.3% for GGR-18 while also remaining inside the expected confidence interval. When we consider the number of decisions needed to arrive at these estimates, LCA with CA-R only required 120 human decisions for a database of 468 ground-truth IDs (0.26 decisions per ID). Furthermore, the 120 human decisions were in addition to 13,307 automated reviews, indicating a decision automation rate of 99.1%. Using LCA with the quality baseline annotations results in 420 human decisions against 22,552 automated decisions (98.2% automated). In contrast, the baseline Graph ID algorithm with the

baseline annotations approximated the population estimate with 2,414 human decisions for 487 IDs (4.96 decisions per ID). That result was generated with only 12,988 requested automated reviews, an automation rate of 84.3%. Likewise, the Graph ID simulation with CA-R had 642 human decisions out of 5,553 total decisions (88.4% automated). These results indicate that using CA-R increases the automation of ID curation and dramatically reduces the number of automated reviews that are needed.

The next chapter will reveal the details of the Great Grévy's Rally and the procedure used in 2016 and 2018 to provide a population estimate. The original process used for those photographic censusing events did not include Census Annotations, Census Annotation Regions, or LCA. However, a culminating experiment is provided that shows the intended use case and most up-to-date procedure.

CHAPTER 6
PHOTOGRAPHIC CENSUSING OF GRÉVY'S ZEBRA IN KENYA

The Grévy's zebra (*Equus grevyi*) has been the focus of active population monitoring efforts in Kenya [241], [246] because of its *Endangered* status and shrinking population [1]. However, previous monitoring efforts [378], [379] were limited to small portions of the total population or did not attempt to build a comprehensive animal ID database with every individual. In contrast, a census of the entire species would provide ecologists with an unprecedented level of insight into how conservation efforts are impacting the Grévy's population. Furthermore, a repeat census on the same population could help establish useful ecological trends and, hopefully, chronicle the species' return to sustainability. To this end, the previous chapters have demonstrated the automated algorithms (the detection pipeline in Chapter 3 and Census Annotation in Chapter 5) and procedures (components in Chapter 4) that are needed to perform a large-scale photographic census over time. These tools represent a paradigm shift in animal population monitoring and culminate nearly a decade of academic research in applied computer vision methods and on-the-ground data collection.

This chapter presents the analysis of the Great Grévy's Rally (GGR), a large-scale photographic census of the entire Grévy's zebra population in Kenya. The GGR process is offered as an improved successor to the prototype process established during the Great Zebra and Giraffe Count (GZGC) [2]. The GZGC was held March 1-2, 2015 at the Nairobi National Park in Nairobi, Kenya, and was organized to estimate the resident populations of Masai giraffes (*Giraffa camelopardalis tippelskirchi*) and Plains zebras (*Equus quagga*) within the park. In contrast, the GGR was first held on January 30-31, 2016 in the Laikipia region of central and northern Kenya, covering the known range of Grévy's zebra within the country. The GGR was repeated on January 27-28, 2018 for the same survey area and added a second census on reticulated giraffes (*Giraffa reticulata*). The reticulated giraffe population has an overlapping resident area with Grévy's zebra in Kenya, making it an ideal species for simultaneous photographic censusing. Figure 6.1 provides a map of Kenya and the respective survey areas for each of the three censusing rallies. We can see that the area covered by the GGR events is much larger than the GZGC, incorporates the conservation areas of

Portions of this chapter previously appeared as: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

Portions of this chapter previously appeared as: J. Parham, C. Stewart, T. Berger-Wolf, D. Rubenstein, and J. Holmberg, "The Great Grevy's Rally: A review on procedure," in *AI Wildlife Conserv. Workshop*, Stockholm, Sweden, Jul. 2018, pp.1–3.

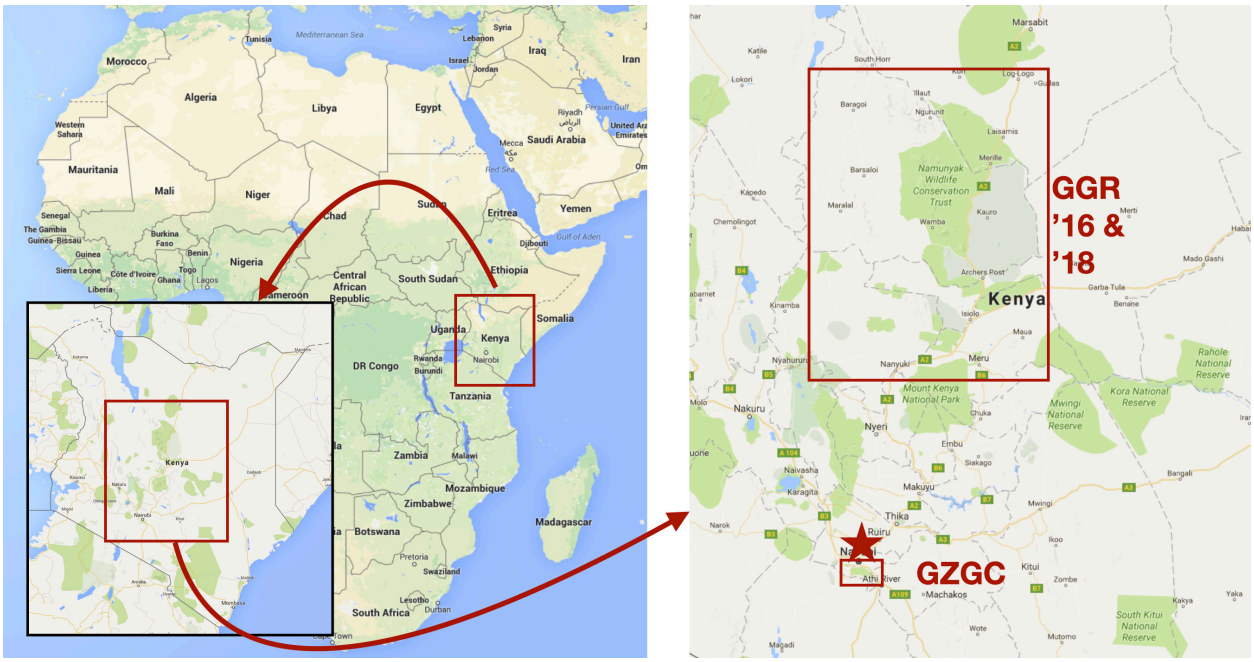


Figure 6.1: The map of the survey boundaries for the GZGC, GGR-16, and GGR-18 photographic censusing rallies. The survey area of the GZGC was confined to the Nairobi National Park in Nairobi, Kenya, and censused Masai giraffes and Plains zebras. The capital city of Kenya, Nairobi, is represented with a red star. The Great Grévy's Rally 2016 and 2018 took place in the northern Laikipia region of Kenya, the primary residence area of Grévy's zebra and reticulated giraffe. Rendered with Google Maps. Best viewed in color.

multiple Kenyan counties, and is concerned with an open animal population.

The GGR events in 2016 and 2018 (referred to as "GGR-16" and "GGR-18", respectively) significantly refined the photographic censusing process used during the GZGC. Both GGR-16 and GGR-18 sampled a significantly larger geographical area, produced more confident population estimates than historical estimates, and massively increased the amount of automation with better computer vision algorithms. Across the three censusing rallies, approximately 100,000 photographs were processed and collected by more than 400 volunteer citizen scientists, including biologists, park rangers, computer programmers, tourists, and school children. As a result, the GGR is the largest known photographic census of Grévy's zebra ever performed and is estimated to have cataloged 70% of all Grévy's zebra in Kenya (as we will see later), representing the most accurate and comprehensive census of the species to date. Furthermore, the Grévy's zebra population estimates from the GGR-18 have been accepted by the Kenyan government as the country's official population count. This recognition has never before been granted to a non-governmental group.

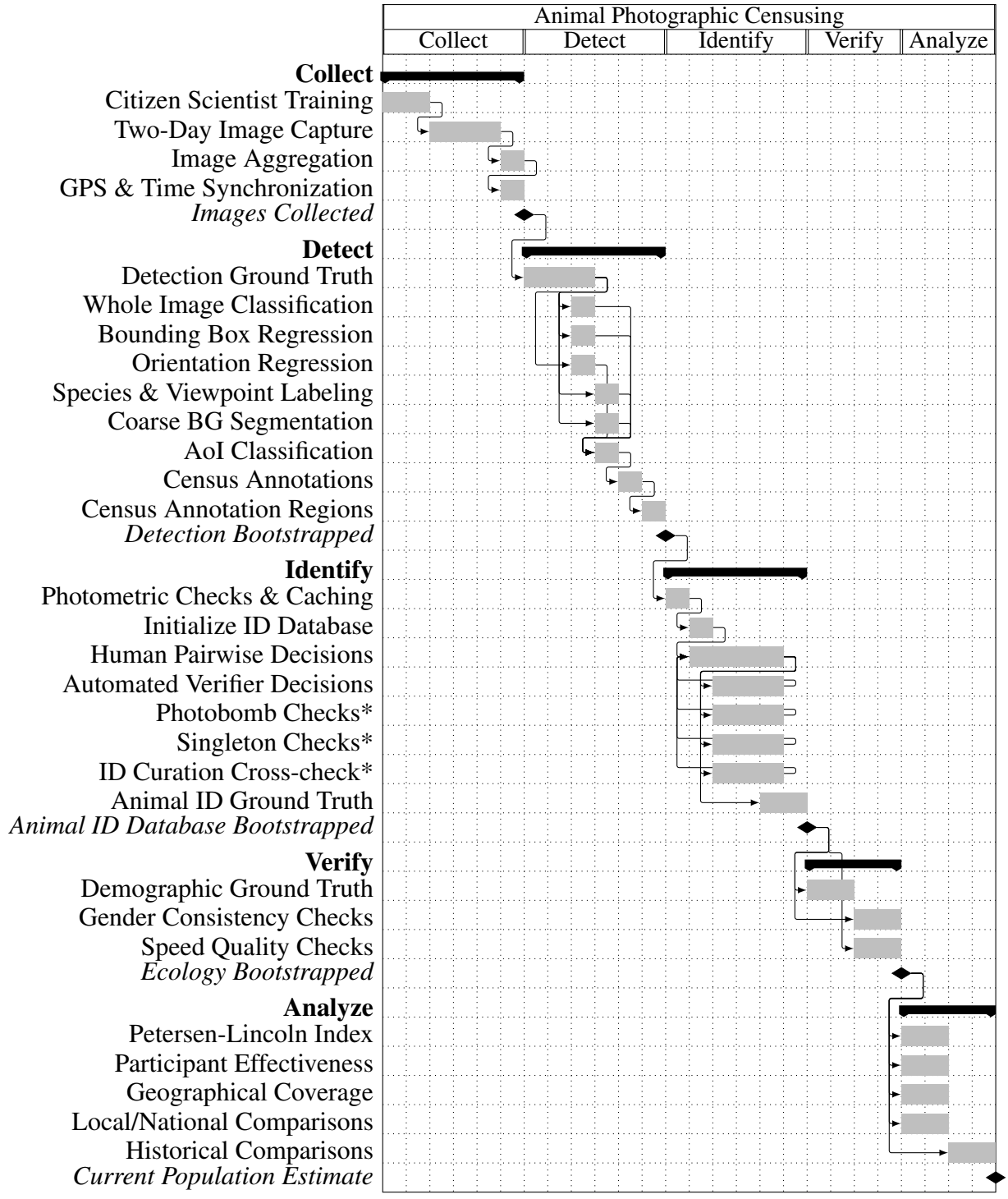


Figure 6.2: A Gantt chart for the recommended process used for animal photographic censusing, including data collection and boostrapping of the detection pipeline for novel species. *Steps were not used during the GGR-16 and GGR-18.

Table 6.1: The number of cars, cameras, and photographs for the GZCD, GGR-16, and GGR-18 photographic censusing rallies. The GGR rallies had over three times as many citizen scientists who contributed four times the number of photographs for processing. The GGR-18 rally, as compared to GGR-16, included a 33% increase in photographers and a 21% increase in the number of photographs collected. [GZGC & GGR-16] ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44. [GGR-16 & GGR-18] ©2018 IJCAI. Reprinted, with permission, from: J. Parham, C. Stewart, T. Berger-Wolf, D. Rubenstein, and J. Holmberg, "The Great Grevy's Rally: A review on procedure," in *AI Wildlife Conserv. Workshop*, Stockholm, Sweden, Jul. 2018, pp.1–3.

	Cars	Cameras	Photographs
GZGC	27	55	9,406
GGR-16	121	162	40,810
GGR-18	143	214	49,526

In summary, this chapter has two goals. The first is to describe the procedures of the GGR data collection events and processing, contrasting them with the earlier GZGC event. This process includes using the detection pipeline (without Census Annotations) and the Graph ID animal ID curation algorithm to build an ID database, which was the current state of the analysis at the time in 2016 and 2018. As described below, this database is checked with various automated tools and human decisions for redundancy and subjected to quality checks to ensure accuracy. Furthermore, the final population estimates for both rallies are generated after extensive human effort because, as stated, some of the automation tools presented earlier in this dissertation were unavailable at the time. The second goal of this chapter is to re-analyze the GGR-18 event using these new tools. This new analysis is a culminating experiment to show the effectiveness of the methods developed (CA and CA-R) or incorporated (LCA) to reduce human interaction while maintaining a consistent population estimate. The updated mathematical framework (see Section 4.3) is also used for the first time to fine-tune the final population estimate after taking into account the overall estimated effect of the machine learning algorithm's errors.

6.1 The Great Grévy's Rally (GGR) in 2016 and 2018

Both Great Grévy's Rally censusing events followed the same procedure, presented as a complete process flow diagram in Figure 6.2. One of the primary benefits is that all of the machine learning components can be bootstrapped as the analysis proceeds with the help of human annotators. This design is crucial as ready-to-go, pre-existing machine learning models are not available for most endangered species. Furthermore, once the machine learning components are successfully trained, they can be reused for future events with the same collection procedure and species of interest. This section provides details for how images were collected and aggregated for processing during the GGR events and describes how the detection pipeline was applied on real-world imagery. First, images collected from citizen scientists need to be aggregated and synchronized for accurate GPS and time metadata. Next, the resulting annotations must be curated into an animal ID database with ranking and verification algorithms. The following discussion explains how the GGR-16 and GGR-18 events created their respective ID databases, which were subject to different automated algorithm versions at the time. Lastly, after consistency checks, the population estimates for Grévy's zebra and reticulated giraffes are reported.

6.1.1 Image Collection with Citizen Scientists

A vital feature of the censusing procedure is the ability to distribute and parallelize the collection of animal imagery. The robustness of the sight-resight study critically depends on capturing as many sightings and resightings of individuals. This design is in sharp contrast to a count-based estimate which must always be careful to avoid double counting and overlapping sample regions. Additional advantages of using cameras during a census include 1) it provides actionable evidence of where a specific individual was in time and location (which allows for the possibility of future auditing) and 2) the mechanism is easy to teach to the average person. Furthermore, by not requiring specialized hardware – only a car and a GPS-enabled camera – a large area can be surveyed efficiently, with many photographers overlapping the same geographical area at the same time. Another essential feature of the data collection is its cost-effectiveness, as citizen scientists, volunteers, tourists, field guides, school children, park rangers, scientists, and any other stationary ground-based sources (e.g., camera traps) can all volunteer to contribute data.

The number of cars and volunteers, and the number of photographs taken for the three rallies, are summarized in Table 6.1. Since the volunteers taking photographs are *mobile*, they can go where the animals are; this is in stark contrast to data capture that uses only static camera traps or

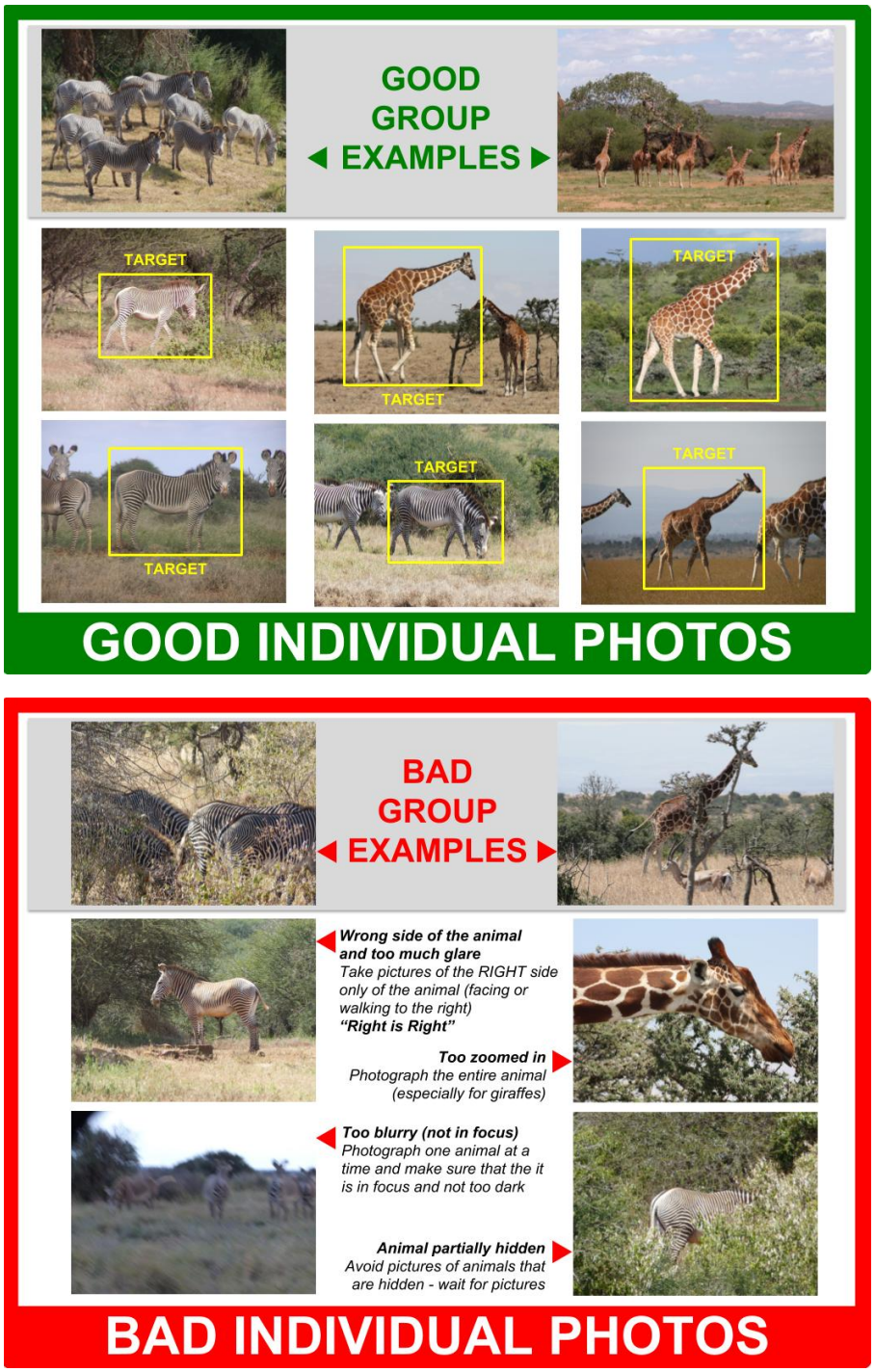


Figure 6.3: An image of the participant training "cheat sheet" used during the GGR-18 photographic censusing rally, showing the *do's* and *don'ts* for capturing images. The examples are explicitly updated to bias the participants towards taking better pictures of AoIs, even showing a primary *target* on the good examples.

fixed-route surveys. It is worth noting that the number of images collected during the GGR-18 is 20% higher and with 30% more photographers compared to the GGR-16 event. Furthermore, it was recognized that the volunteer photographers for all three censusing events did not expect to be paid for their effort. In other words, there seemed to be an intrinsic worth for participants to be part of a scientific endeavor, which was enough compensation in itself. This fortuitous effect suggests that photographic censusing is an effective method of community engagement.[25] The GGR-18 event was able to recruit photographers that participated in the GGR-16 event, exemplifying the fact that some participants were willing to volunteer their time without the need for incentives from the organizers. Before we delve deeper into the analysis of the collected images, let us review the participants' training procedure that was provided prior to each event.

6.1.1.1 Citizen Scientist Training

The participating photographers were asked to go into an assigned survey area via car and capture images of the species of interest (i.e., Grévy's zebra and reticulated giraffe). Prior to departing, each participant was given a training document and a Cheat Sheet (Figure 6.3) that showed the common *do's* and *don'ts* of a photographic collection. For example, the participant training document given to participants of the GGR-18 is provided in Appendix A, and it includes instructions on how to set up the Nikon GPS-enabled cameras. The participant instructions were updated slightly between the GZGC, GGR-16, and GGR-18 censusing events based on a better understanding of how the detection pipeline and ID systems were failing. The training process created a feedback loop, where subtle differences in the training instructions influenced the quality of images collected. For example, the GGR-18 training instructions encouraged the photographer to:

1. Pick a single subject (of the target species and target viewpoint),
2. Always take a picture of an animal in the foreground,
3. Place the subject in the center of the photograph, and
4. Zoom the camera such that the animal covers around 50% of the image (if possible).

All photographers for the GZGC were requested to take pictures of the left sides of plains zebras and Masai giraffes, while photographers for the GGR were requested to take pictures of

[25] Refer to [2] for an example on how to reward volunteer citizen scientist participants for their time and image contributions. For GZGC, a same-day print-out was provided to each participant that listed known and new animals they photographed.

the right sides of Grévy's zebras and reticulated giraffes. Having a consistent viewpoint (left or right) allows for more effective sight-resight and reduces the chance of ML errors. Furthermore, the distinguishing visual markings for these species are not left-right symmetrical, so the animal's appearance differs (sometimes significantly) from side to side. This asymmetry means that a right-only census must discard a left viewpoint sighting as irrelevant data.

Along with guidance on species, photographers were shown examples of good/poor quality photographs emphasizing 1) the correct side of the animal, 2) getting a large enough and clear view, and 3) seeing the animal in relative isolation from other animals. To better guarantee a valuable sighting, GGR photographers were requested to take about three pictures of the right side of each Grévy's zebra they saw. In both the GZGC and GGR, photographers were invited to take other pictures once they had adequately photographed each encountered animal, causing miscellaneous photographs to be collected. This decision was primarily to help minimize photographer fatigue and allow flexibility when an exciting or otherwise rare species was encountered.

In contrast, the instructions for the GGR-16 did not focus on a target animal, only emphasizing the correct species and viewpoint. The above changes nudged the photographers during the GGR to take better pictures of animals by implicitly focusing on the concepts that overlap with Annotation of Interest (AoI, see Section 3.6). To make the *a priori* decision more straightforward for the AoI detection component, specific examples and instructions to guide participants into taking better images were provided. Furthermore, the concept of a Census Annotation did not exist when the GGR-16 or GGR-18 collections were performed. However, the benefit of asking photographers to focus on features consistent with AoIs is that it also biased the participant to capture good CA examples. We saw in Section 5.1.1 a strong correlation between AoI and CA, indicating that the collected animal sightings were still strongly biased towards identifiability.

6.1.1.2 GPS Cameras & Time Synchronization

Upon registering for the GGR-18 censusing rally, all participants were given a GPS-enabled camera and a paper "camera card". This procedure is similar to the GZGC, except for how GPS locations were synchronized across cars. During the GZGC, a dedicated GPS dongle was provided for each car, and participants could bring their cameras. Unfortunately, this open policy proved to be a synchronization challenge across multiple timestamp formats, failures to start GPS recording, and other miscellaneous inconsistencies or problems. As a result, the GGR-16 and GGR-18 procedures were improved to address these issues:

Figure 6.4: An image of the camera card used for the GGR-18 participant "photographer 1" who was assigned to "car 1". A QR detection algorithm was used to automatically localize the first photograph that was used to sync all participants in a car.

1. A Nikon GPS-enabled camera was provided to every car that volunteered to take pictures for the rally. This camera is always labeled with the letter "A" within the car. In addition, a GPS camera replaced the GPS dongle that was provided to every car during the GZGC. All photographers' times of their photographs were assigned locations via a look-up table from the GPS log.

2. A QR code was added to the camera card that provided a link to the Great Grévy's Rally website[26], which also embedded the car number and photographer letter into the URL.

3. The "3-2-1 Snap" handout used during the GZGC[27] was combined and consolidated with the camera cards. The for GZGC, all participants in a car used a physical sheet of paper to take a synchronized photograph at the start of the day. Each participant then wrote the local time for when the picture was taken on their registration cards. The written time, the photograph,

[26] `greatgrevysrally.com` (Accessed: Oct. 29, 2021).
[27] See Section 2.1 of [2], mentioned there as a photographer's $Image_0$.

and the camera's internal clock were used to calculate the correct time for each image a photographer contributed. For the GGR-16 and GGR-18, the "3-2-1 Snap" wording was added to the front side of the camera card, which was used to coordinate all photographers in a car. Since at least one camera in the car was guaranteed to support GPS (and therefore has access to accurate timestamps), participants did not need to write down the local time.

An example sync image of the camera card with QR code can be seen in Figure 6.4. During processing, the images were automatically scanned to find the QR code for each photographer in a car. The timestamps of the QR code are associated with the correct timestamp provided by the GPS camera, which receives accurate date and time data from the orbiting satellites. A given photographer's images were then assigned a "timedelta" (i.e., a time offset), which was used to correct their respective EXIF timestamps to local Kenya time. Furthermore, there is a unique QR code for day 1 and day 2 of the census rally. The second QR code and "3-2-1 Snap" image are used to cross-reference and check the *timedelta* calculation for a given photographer. The benefit of using a separate QR code for days 1 and 2 is that it adds redundancy because some photographers forgot to take the image on either day. In that event, the assumption is that the system clock of each participant's camera is at least internally consistent, so one timestamp is sufficient to establish the correction factor.

6.1.1.3 Aggregating Multiple Cameras

After the photographers took the images, censusing rally staff collected and stored them onto a single centralized computer. Each photographer's camera card was used to create a named folder of that participant's images during collection (e.g., "GGR-18/CAR-1/CAMERA-A"). The photographers within the same car had the same car number, each with their own unique camera letter. The letter "A" was reserved for the census-provided GPS-enabled camera in the car taking photographs. Unfortunately, this relatively simple procedure still resulted in the inappropriate images being grouped – the approximately 250GB of collected data during each event needed to be cleaned and restructured.

The QR detection algorithm searched a photographer's contributed images to find the *first* image that had a QR code. Some data organizing errors were found by comparing the QR camera card photograph (which embeds photographer information) with its assigned folder name. For example, with the GGR-18 data, 45 manual resolutions needed to be made, including renaming folders, merging two folders, and moving folders from one car to another. The first QR image of the

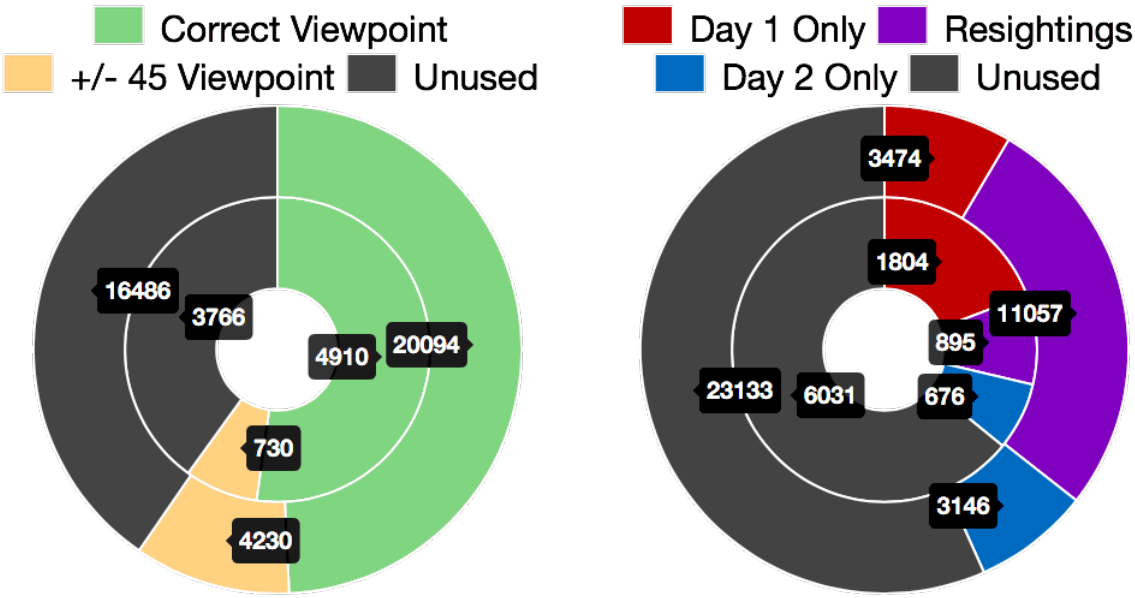


Figure 6.5: The number of photographs (left) that adhered to the collection protocol for the GZGC (inner-ring) and the GGR-16 (outer-ring). The number of photographs that adhered to the viewpoint collection protocol was around 50% (green) for the GGR-16 and the GZGC. The number of which photographs (right) that had sightings on day 1 only, day 2 only, and resightings for the GZGC (inner-ring) and GGR-16 (outer-ring); the sightings data and its colors are meant to mirror that of Figure 6.6. Note that any photographs with no sightings are grouped with unused. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

rally was assumed to have been taken simultaneously with the A camera, providing a method to establish the non-A camera's time offset from local time and approximate GPS location. However, the QR code detection was not perfect. Sometimes a human reviewer had to manually search for the QR code by hand, starting with the photographer's earliest images working forwards in time. Images taken outside of the time range of the two-day event were discarded to preserve the privacy of the contributors. For example, the data collected during the GGR-18 had 53,193 images, but 3,649 were taken outside the two-day time window or geofence boundary of the censusing rally.

6.1.1.4 Adherence to Training Instructions

Now that the images have been collected from the volunteer photographers, we wish to analyze how well the resulting images conform to the provided training instructions. Fir the GZGC and

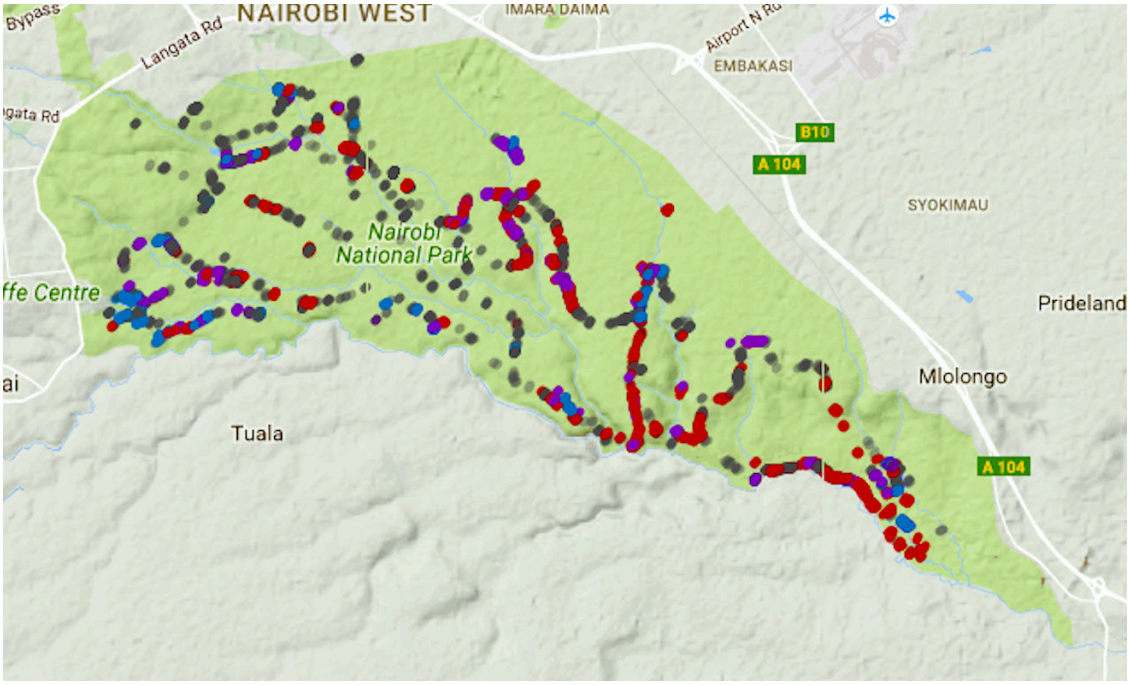


Figure 6.6: The map of image GPS locations from the GZGC censusing rally. Colored dots indicate sightings during the two days of each census; red was from day 1 only, blue was day 2 only, purple was resightings, and gray were unused. Rendered with Google Maps. Best viewed in color. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

GGR-16, all annotations were created by hand, and viewpoints were added to determine how well the citizen scientists followed the data collection protocols. As discussed earlier, citizen scientists were instructed first to take photographs from specific viewpoints on the animals – left side during the GZGC and right sides for Grévy's zebras (GGR-16) – and then take additional photographs if they desired. As such, the distribution of viewpoints is a strong indicator of adherence to the protocol. For example, Figure 6.5 (left) shows that around 50% of the photographs in the GZGC and GGR-16 had an annotation from the desired viewpoint (green). Furthermore, when the photographs of neighboring viewpoints (yellow) are taken into account, the percentage grows to 60%. The graph in Figure 6.9 reinforces the argument of good adherence, showing how the photographs were used during the analysis. The most significant percentage of photographs filtered out did not include animals of the desired species. The second highest percentage was from poor photograph quality. Even so, the number of photographs used is still around 50% for the GGR-16.

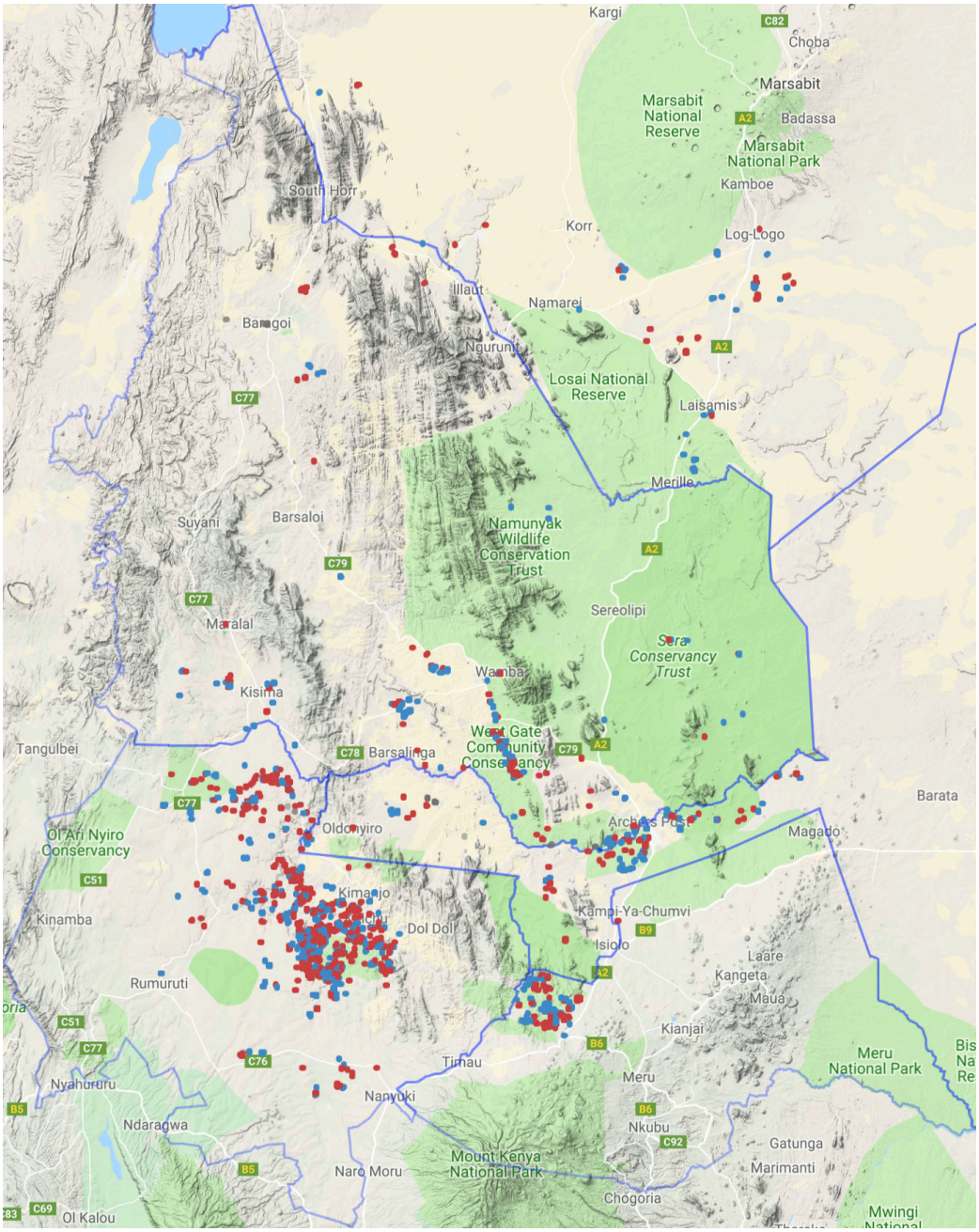


Figure 6.7: The map of image GPS locations from the GGR-16 censusing rally. Colored dots indicate sightings during the two days of each census; red were from day 1 only, blue were day 2 only, purple were resightings, and gray were unused. The blue area lines indicate Kenyan county boundaries. Rendered with Google Maps. Best viewed in color. ©2018 IJCAI. Reprinted, with permission, from: J. Parham, C. Stewart, T. Berger-Wolf, D. Rubenstein, and J. Holmberg, "The Great Grevy's Rally: A review on procedure," in *AI Wildlife Conserv. Workshop*, Stockholm, Sweden, Jul. 2018, pp.1–3.

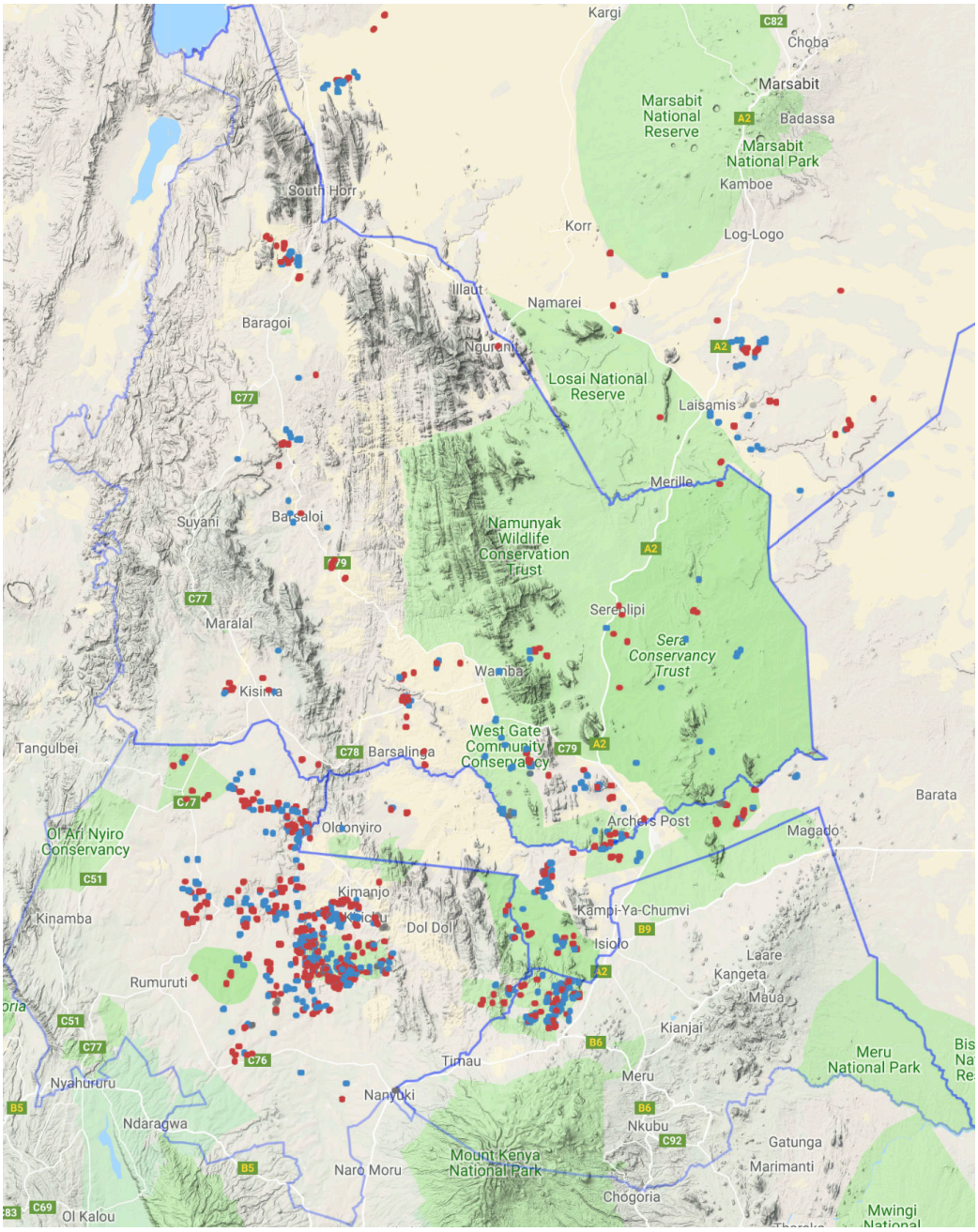


Figure 6.8: The map of image GPS locations from the GGR-18 censusing rally. Colored dots indicate sightings during the two days of each census; red were from day 1 only, blue were day 2 only, purple were resightings, and gray were unused. The blue area lines indicate Kenyan county boundaries. Rendered with Google Maps. Best viewed in color. ©2018 IJCAI. Reprinted, with permission, from: J. Parham, C. Stewart, T. Berger-Wolf, D. Rubenstein, and J. Holmberg, "The Great Grevy's Rally: A review on procedure," in *AI Wildlife Conserv. Workshop*, Stockholm, Sweden, Jul. 2018, pp.1–3.

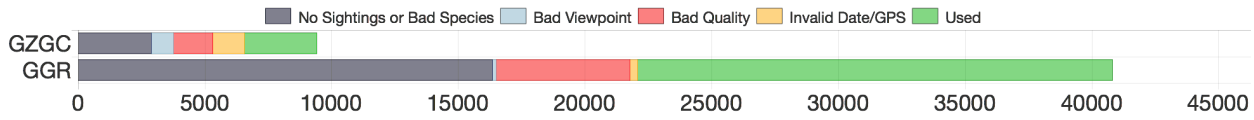


Figure 6.9: The numbers of collected photographs from the GZGC and GGR-16 and how they were used. A large number (gray) were filtered out simply because they had no sightings or captured distracting species. We further filtered the photographs taken of undesired viewpoints and had poor quality. Lastly, we filtered photographs that were not taken during the two days of each rally (some volunteers brought their cameras with non-empty personal memory cards) or had corrupt/invalid GPS. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

6.1.1.5 Geographic Coverage & Image Distributions

Despite a skewed distribution, there was still strong area coverage across all three census rallies. The locations of images taken during the GZGC can be seen in Figure 6.6, GGR-16 in Figure 6.7, and GGR-18 in Figure 6.8. Note that the maps shown in the figures are at vastly different scales (refer to Figure 6.1), and the coverage plots in the GZGC essentially show the roads through the Nairobi National Park. The park was split into five zones [2] to help enforce coverage, which was very good in most cases. For the GGR, the 25,000 km^2 survey area was broken into 45 counting blocks with variation in the animal density due to the presence of human settlements and the availability of habitat and resources to sustain Grévy's zebras. These 45 blocks (comprised mainly of protected conservation areas) were further organized into one of 5 Kenyan counties covering the survey area: Isiolo, Laikipia, Marsabit, Meru, and Samburu. The blue lines on the GGR maps show county boundaries. The spatial distributions of resightings are pretty uniform for both rallies, indicating that the respective counting block partitioning schemes accomplished their intended goals. Furthermore, there were also five zones for the GGR events to ensure geographically isolated areas were properly sampled (see the top of Figure 1 in [380]).

Figure 6.10 plots the distribution of photographs per camera. We see that some photographers and cars are more prolific than others. For example, the car that contributed the most images was nearly twice as productive as the second-most-productive car during the GGR-16 and produced nearly 3.5 times as many images as the most active car during the GZGC. There are several possible reasons for the observed drop-off. In the GZGC, while some photographers were professional ecologists and conservationists, others were volunteers recruited on-site. Therefore, a significant

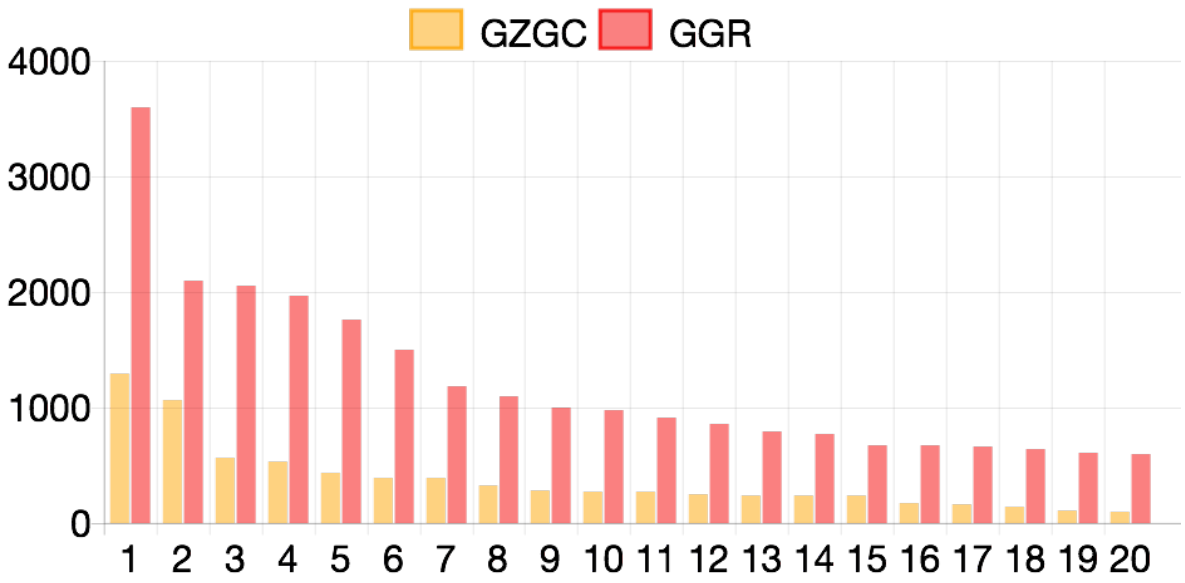


Figure 6.10: The number of photographs taken by the top 20 cars during the GZGC and the GGR-16. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

difference in the commitment to take large quantities of photographs was expected. For GGR-16, where volunteers were recruited in advance, the expertise was more uniform, but each car also had an assigned region, and the regions differed significantly in the expected density of animals. A similar distribution was seen with the data collection for GGR-18, with some cars contributing a significant percentage of the photographs. The key insight is that these high-volume photographers also sighted many individuals in the population and were assigned to areas with a known dense population. As seen on the coverage maps, the areas with the most population also have the highest density of photographs.

One of the known issues with the GGR-16 censusing rally was the poor coverage in the northern areas of the overall survey area. During the GGR-18, additional participants were assigned explicitly to the northern areas, and the estimate essentially "recovered" approximately 400 animals that were missed during the first censusing rally. Figure 6.11 shows a heat-map for the locations of identified zebras during the GGR-16 and GGR-18. The red line indicates the delineation between the southern and northern blocks, showing that the GGR-18 census had a much more thorough coverage. The county-by-county population statistics (which is shown in Table 6.3) for the GGR-16

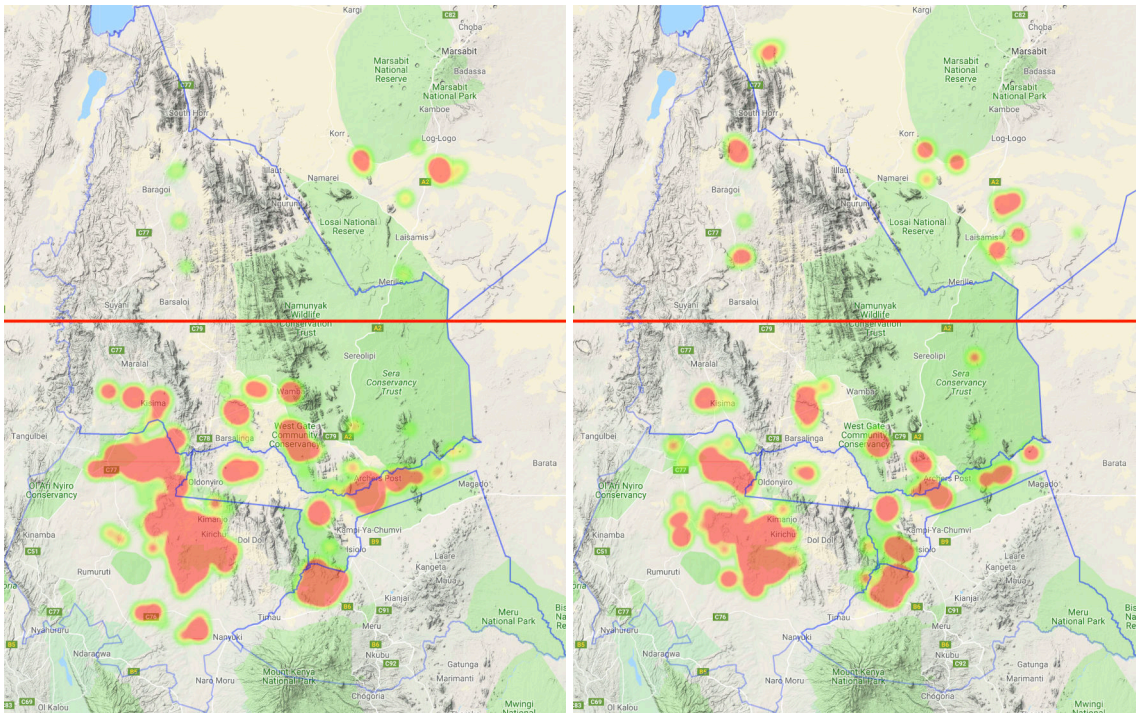


Figure 6.11: A heatmap of identified animals captured during the GGR-16 and GGR-18 photographic censusing rallies. The coverage in the northern blocks (above the red line) was improved during the GGR-18 (right) as compared to the GGR-16 (left).

and GGR-18 also support this conclusion as the population estimates in the southern counties are stable over the two-year interval. The northern region was simply not sampled thoroughly enough during the GGR-16, resulting in that area violating Assumptions 1 and 4 from Section 4.3.1.

6.1.2 Building the Animal ID Database

The processing of the collected data[28] includes multiple stages and is designed to be a comprehensive pipeline that takes raw images as input and returns a set of named sightings with age and sex metadata. To provide a quick summary, we 1) curated a portion of the data for detection training with multiple reviewers per image, 2) trained the detection pipeline and applied it on all collected images, 3) used HotSpotter to rank all relevant annotations, 4) used the Graph ID algorithm to suggest manual review decisions to an asynchronous web interface or the VAMP automated decision algorithm and 5) asked ecologists to generate age and sex labels for all named individuals.

[28]The GGR-16 and GGR-18 processing pre-dated the development of Census Annotations, Census Annotation Regions, and the LCA graph curation algorithm.

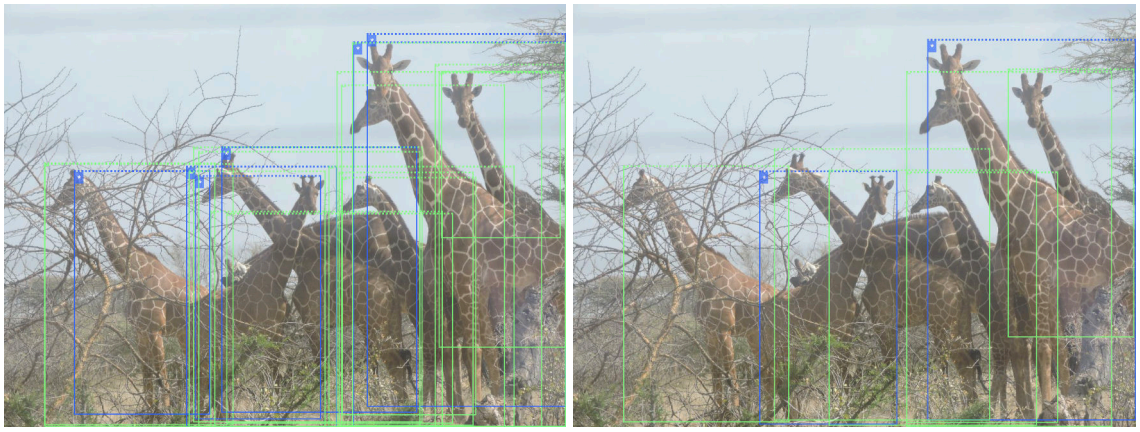


Figure 6.12: An example image of giraffe from the GGR-18 photographic censusing rally, showing the input (left) and output (right) of the triple-marriage assignment problem. Each image from the 10% of annotated GGR-18 data is shown to three independent reviewers. A triple-marriage algorithm is used to merge these into the final candidate bounding boxes (and AoI assignments) that are used for training the localizer.

6.1.2.1 Applying the Detection Pipeline

New Grévy's zebra and reticulated giraffe localization models were explicitly trained for the GGR-18, ignoring pre-existing models from the GGR-16 and GZGC. These models were re-trained due to training process improvements (i.e., an updated Python implementation) and used annotations generated by a more robust bounding box collection methodology. A total of 5,000 images were annotated for the GGR-18, with at least three independent reviewers for each image. A pool of 14 reviewers was asked to annotate bounding boxes for 10% of the dataset during the GGR-18. The triple-reviewer procedure was not used with the GZGC or GGR-16 events for two reasons: 1) in the interest of time, the reviewers' workload was limited to one review per image (and done by hand for all images), and 2) the analysis of the GGR-18 data included the AoI component, which required more reliable and robust ground-truth annotations. Since each image was annotated slightly differently by three individuals (and since AoI decisions are somewhat subjective), these bounding box candidates needed to be merged (or "married") into a set of finalized bounding box candidates. A greedy three-person marriage algorithm compared the bounding boxes across the three reviewers, prioritizing the merging of boxes with the highest joint Intersection Over Union (IoU) percentages and starting with groups of three highly overlapping boxes (one from each reviewer). This process continued until a threshold was met (IoU 25%). After all three-box marriages were assigned, a

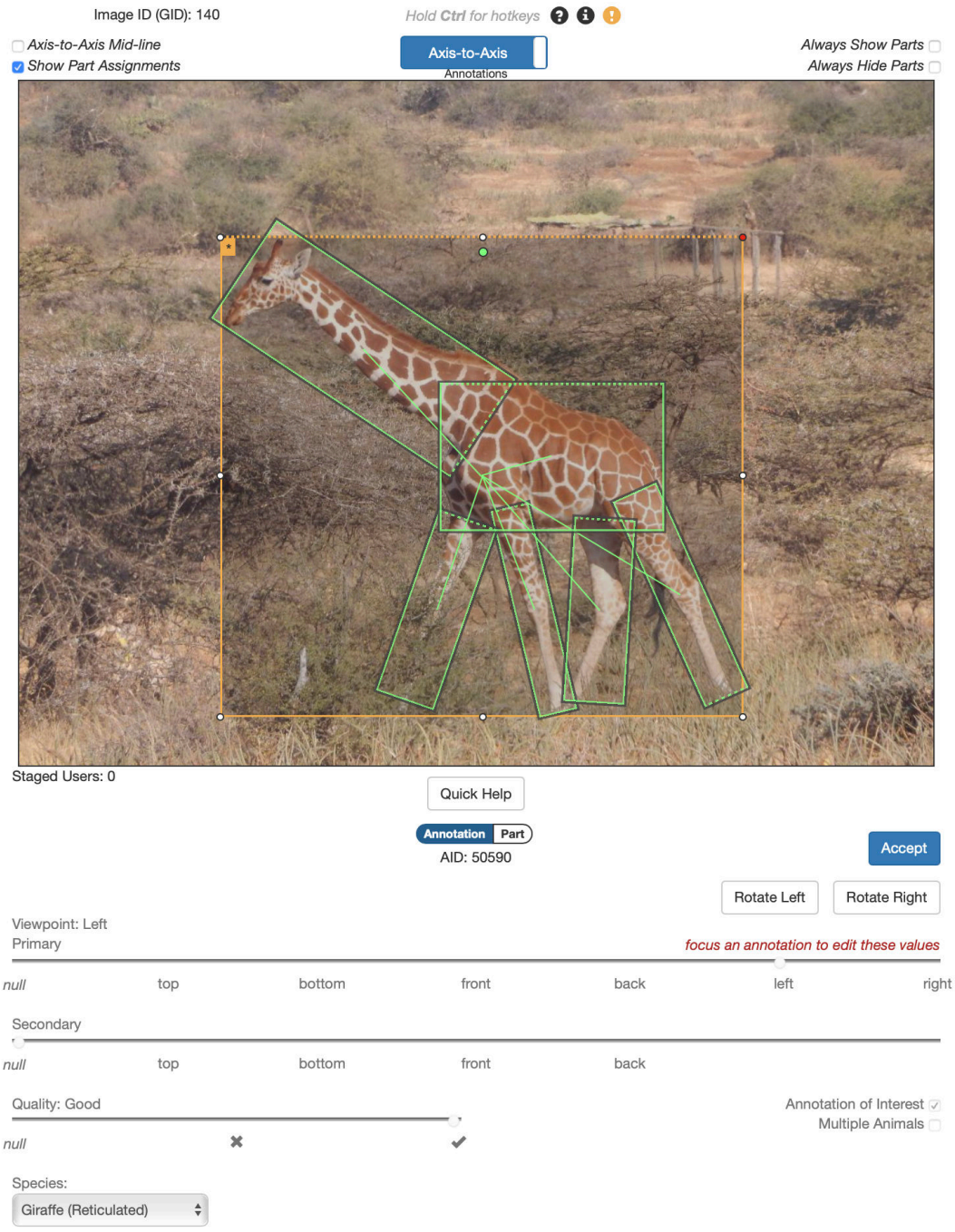


Figure 6.13: An image of the updated web interface for bounding box annotation. The updated interface was rewritten from the ground-up as used to annotate ground-truth data during the GZGC censusing rally. The new interface is responsive, supports annotation parts and metadata, and is released as a public open-source tool.

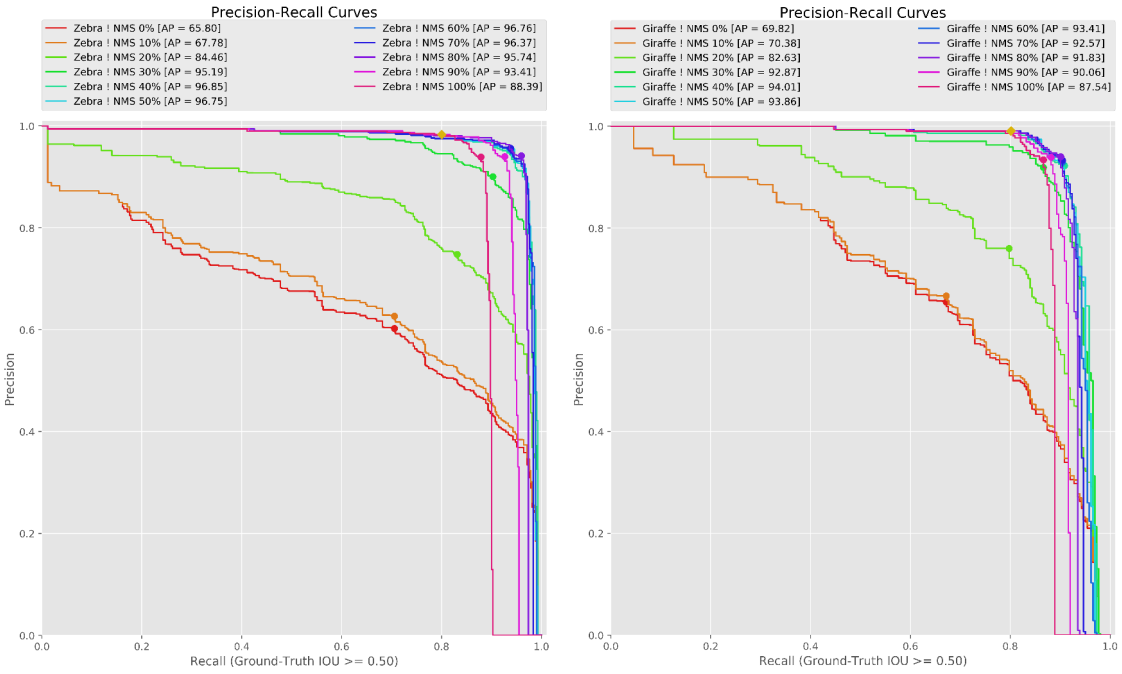


Figure 6.14: The precision-recall performance curves for the localizer during the GGR-18 photographic censusing rally. The performance of the localizer on zebra (left) and giraffe (right) AoIs for the GGR-18. The NMS threshold that achieved the highest precision-recall AP for each species was chosen, followed by the operating point that was the closest to the top-right corner, balancing precision and recall for the highest area of AP. Coincidentally, both species had the best performance with a NMS threshold of 40% overlap and – approximately for both species – a best operating point at 0.4 for the detection confidence.

second round of two-box marriages was performed. An example of a marriage solution can be seen in Figure 6.12. Furthermore, AoI flags were determined independently for each annotation before the marriage assignments. These flags are vital to properly tune the localization models to de-prioritize background animals. Figure 6.13 displays the web interface that the reviewers used to annotate all ground-truth detection data. The interface was further updated since its use in the GZGC to add support for part bounding boxes. The bounding boxes and their AoI assignments (seen in blue) for a given image are displayed together (left), and the final bounding boxes and AoI assignments are also shown (right). The final AoI flag was determined by a majority vote, with married pairs of two being marked as an AoI if at least one reviewer considered their bounding box an AoI.

The whole-image classifier (see Section 3.2) and localizer were trained with an annotated

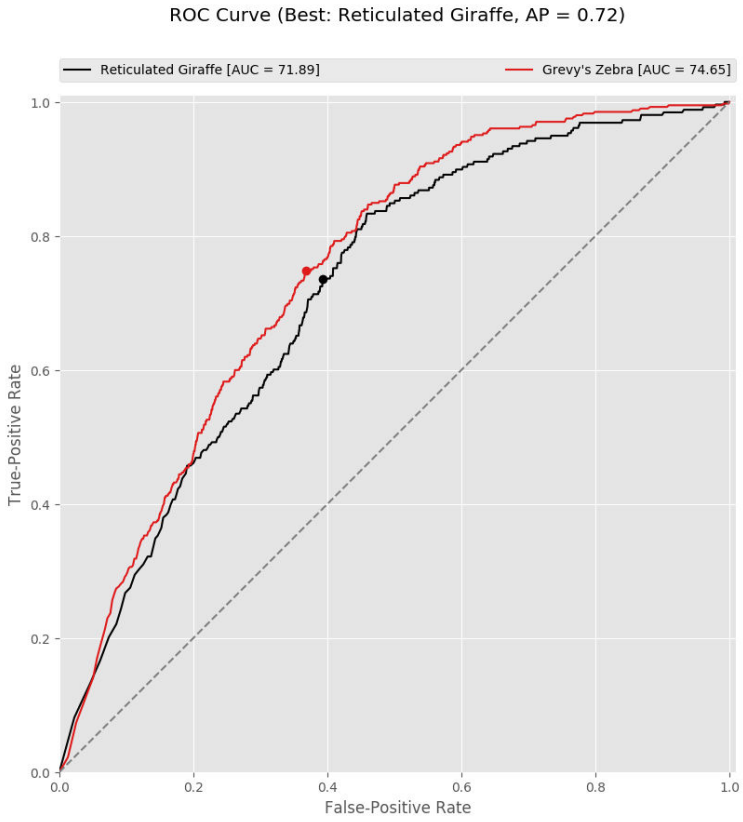


Figure 6.15: The ROC performance curves for the AoI component during the GGR-18 photographic censusing rally. The AoI classifier was trained to predict the majority decided AoI flags on each annotation annotated from the 5,000 training image set for the GGR-18.

subset of bounding boxes with their species labels. The precision-recall performance curves of AoI detections can be seen in Figure 6.14 for Grévy's zebra (left) and reticulated giraffe (right). The figures show varying levels of non-maximum suppression (NMS) applied on the output detections, both achieving an AP of at least 94% for their best respective configurations. The next step was having reviewers add viewpoint assignments to each animal (across the cardinal and sub-cardinal eight directions) for training the annotation classifier. The bounding boxes with species assignments also provided sufficient training data for training the coarse foreground-background segmentation classifier. Lastly, reviewers were asked to annotate the AoIs in the image to provide training data for the AoI classifier. The human-curated data was used to bootstrap all of the core detection pipeline components and was then applied to the remaining 90% of the collected data.

The AoI classifier has a classification accuracy of 76.49% on average between the two species, indicating moderate success in eliminating apparent background sightings. The ROC curves of the

re-trained classifier can be viewed in Figure 6.15. The number of identifiable annotations collected for each species can be seen in Table 6.2. At the time of processing the annotations for the GGR-18, this AoI filtering model was deemed to be inadequate. The failure modes for AoI were significant and varied enough that it suggested a new concept for identifiability was needed. The concept of Census Annotations and Census Annotation Regions (see Chapter 5) were partially motivated by this AoI classification performance during the GGR-18. These new components are meant to be a drop-in replacement for AoI in future censusing events, as will be demonstrated below in Section 6.2.

6.1.2.2 Animal ID Curation

There were three primary challenges with using the Graph ID algorithm for ID curation during the GGR-16 and GGR-18: 1) it was still dependent on large amounts of human effort for review 2) its internal ID ranking process (HotSpotter) took a long time to compute for large databases and 3) it had specific implementation inefficiencies that made it hard to fit into memory. The basic building block of the Graph ID algorithm is the ranking results, without which the curation process is aimless and has quadratic complexity. The algorithm uses the ranked list to prioritize which matches are sent to human reviewers and automated decision algorithms, representing the largest source of pairs during the analysis. Furthermore, the total number of reviews will be influenced by the parameters of the ranking algorithm (e.g., GGR-18 returned the top five matches for each query annotation). Some of the ranked matches are intentionally discarded for being poor spatial candidates; some are discarded for failing to pass a score threshold; some are bi-directional duplicates (i.e., A matched B and B matched A, so one match pair is discarded). For example, for 11,916 Census Annotation Regions and a ranked list configuration that returns the top-10 matches (allowing twice as many potential matches as top-5), the HotSpotter algorithm suggests 67,247 total matches. The resulting ranked list is passed to the VAMP automated verifier, which automatically decides pairs above a scores threshold (as specified by held-out validation data). Any match that falls below the threshold is provided to a human for a decision.

The verifier score threshold is not the only way for reviews to be added to the queue of pairs that need a decision from a human reviewer. When the Graph ID algorithm identifies an inconsistency, it immediately adds additional annotation pairs into the queue to find the problem. Ideally, the top of the human review queue (and the first to be provided to a reviewer) is the match that is actively blocking the ID curation algorithm. Unfortunately, this means that the review

process is restrictively iterative. Worse, the processing between each annotation is non-deterministic, meaning that the Graph ID algorithm could take an indeterminate amount of time before providing the next match to review to a human. Herein lies a dilemma: we wish to 1) provide a match to a reviewer such that the ID curation algorithm can resume[29] and 2) never run out of relevant reviews to provide to the active reviewers. For the GGR-16 and GGR-18 processing, it took a team of around a dozen reviewers working concurrently against an ever-updating set (n=500) of candidate matches to review. Even with advanced and accurate machine learning methods for detection, feature description, ranking, and pairwise verification, it still took months of full-time analysis to review comprehensively.

For each annotation, approximately 3-4 human or automated reviews were required (top-5) for the animal ID database to be consistent. For example, the GGR-18 analysis incorporated 10,044 Grévy's zebra and 4,018 reticulated giraffe annotations and required 35,608 total pairwise reviews. Exactly 18,556 reviews were performed by a human reviewer (52.1%) during GGR-18, which indicates that – optimistically – at least 1.5 human reviews per annotation are required on average. Thus, the process used by the GGR-16 and GGR-18 events does not represent a scalable solution to large-scale photographic censusing. What is needed is 1) an order-of-magnitude reduction in the amount of human involvement in photographic censusing and 2) non-blocking curation algorithms that do not require highly iterative workflows. The GGR-16 and GGR-18 events did not benefit from Census Annotation Regions and LCA at the time, and their respective ID curations were mainly performed by hand. Luckily, as we have seen, the ID databases for both events converged and can be used as a comparative baseline for future algorithm development. The following subsection provides some implementation details on how the memory constraints of the Graph ID algorithm were mitigated. The reader can safely skip to Section 6.1.2.4 to review the next step of the photographic censusing procedure on demographics and quality checks.

6.1.2.3 Implementation Details for Tree-based Graph ID Curation

The selected annotations for ID from the detection pipeline were partitioned into a binary tree (four levels for zebras and three levels for giraffes) to more efficiently control the ID curation process. This structure allowed different parts of the ID database to be worked on simultaneously by the same pool of reviewers. Each leaf of the tree was balanced such that there existed roughly 1,000 annotations, with annotations taken by the same car automatically grouped. A reviewer was then

[29] Refer to [13] for specific details on Graph ID and its phase-based workflow.

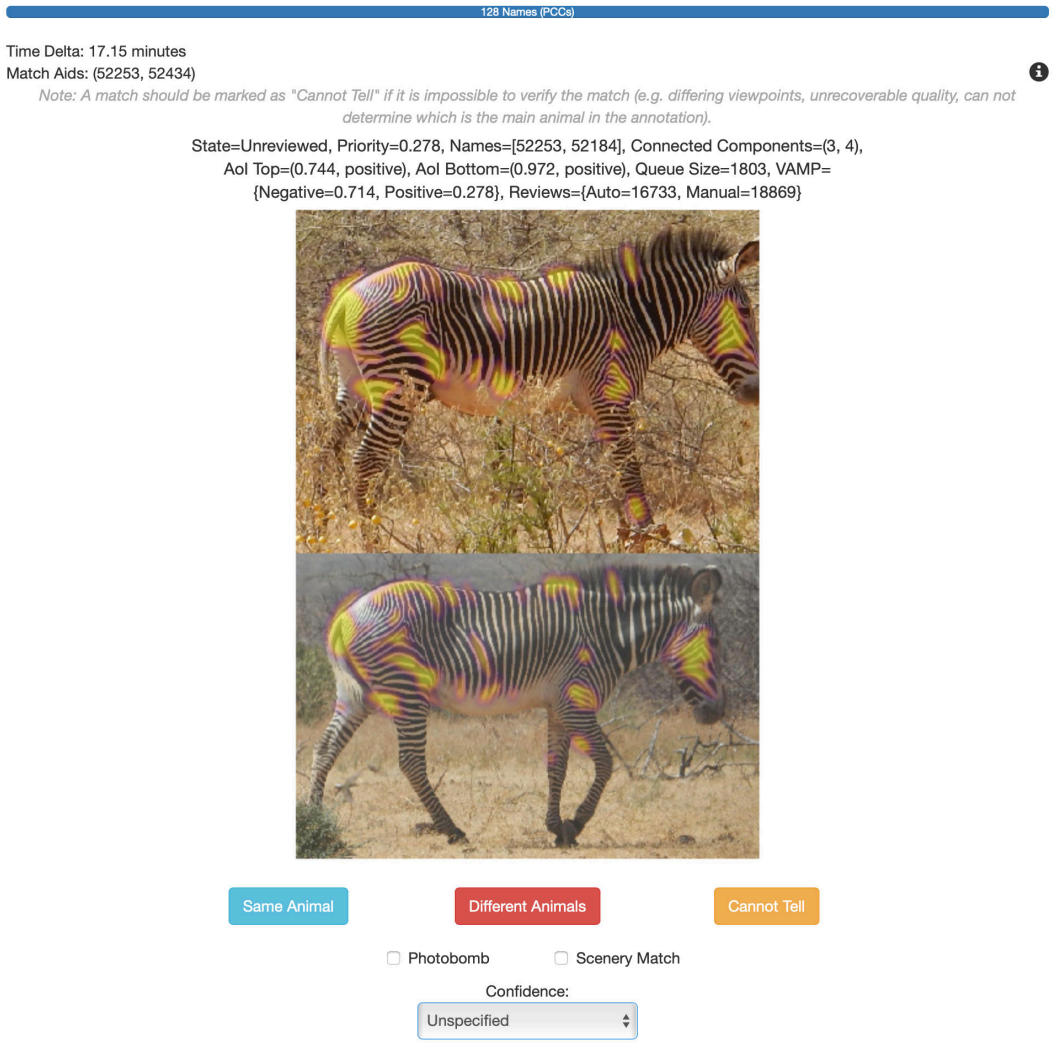


Figure 6.16: An image of the web interface for reviewing matched annotation pairs. The Graph ID algorithm suggests an iterative list of matches for review by humans. We extend the base algorithm to make it asynchronous and allow multiple web-based reviewers to make decisions concurrently. This match shows an example of a `negative` match.

dispatched to work on one of the 16 (or 8) leaves to review whatever matches were ready for human review. If the review queue for a given leaf was empty (e.g., when processing in the background to generate new reviews), we provided the reviewer with a different leaf waiting for a human decision. The interface (Figure 6.16) provides the reviewer with two animal sightings (top and bottom) and a heat-map for the suggested correspondences. Once all of the leaves for a given level had converged, they were merged in pairs of two. The process was then restarted, comparing leaves that were twice as large as the previous round. The curation continued and worked up the binary tree towards the

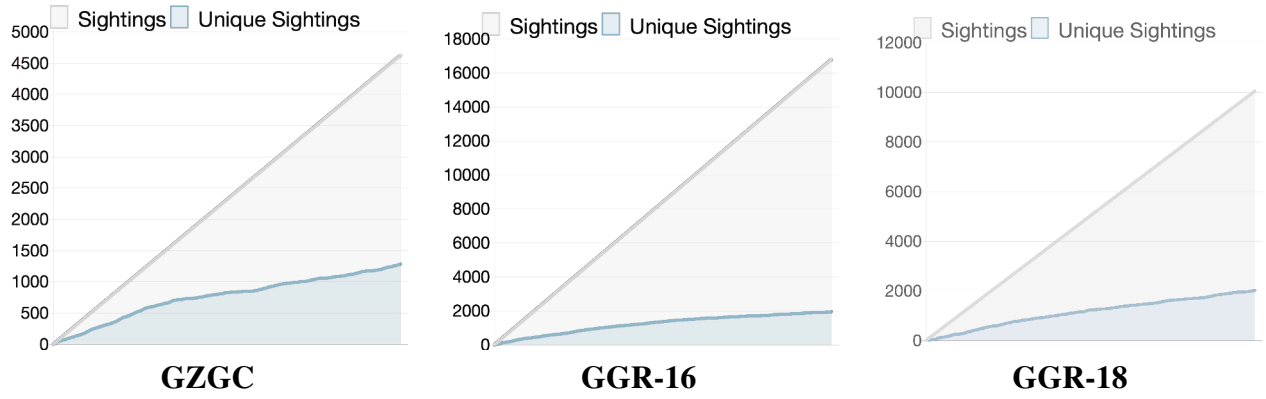


Figure 6.17: A plot of the identification convergence rates for the GZGC, GGR-16, and GGR-18 photographic censusing rallies. The convergence of the identification algorithm during the GZGC [2] (left), the GGR-16 (middle), and the GGR-18 (right). The x-axis shows all collected photographs in chronological order and the y-axis shows the number of sightings against new sightings. The x-axis is the same scale as the y-axis. As photos are processed over time, the rate of new sightings decreases. The smaller slope of the GGR rallies indicate that the rate of resightings for the GGR censusing events were higher than the GZGC. [GZGC & GGR-16] ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

root, which contained all the annotations. As the final matching was performed on the root, the rate of finding new animals slowed. By the time the last two leaves were merged into a single root, the vast majority of the reviews had already been completed. The analysis was terminated when there were no more merges and split cases after a specified period.

6.1.2.4 Demographics & Quality Checks

Once the ID curation was complete, ecologists were asked to manually annotate age and sex information for each animal ID in the database. The reviewer was presented with all of the annotations for a given individual animal to make a more accurate decision (e.g., to browse for unambiguous photographs of genitalia). The review of age and sex also allows for an error checking method, where any cross-gender or cross-age matching mistakes were flagged and corrected. The reader is referred to the field reports [356], [357], [381] for a breakdown on the demographics during the GGR-16 and GGR-18 along with discussion on population stability.

Another name-based check is to ensure travel constraints through GPS and time EXIF

Table 6.2: The number of annotations, matched individuals, and the final mark-recapture population size estimates for the three species of GZGC, GGR-16, and GGR-18. The Lincoln-Petersen (L-P) estimates are calculated with a 95% confidence interval. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

Censusing Rally	Annotations	Individuals	L-P Estimate
GZGC Masai	466	103	119±4
GZGC Plains	4,545	1,258	2,307±366
GGR-16 Grévy's	16,866	1,943	2,269±95
GGR-18 Grévy's	10,044	1,972	2,812±171
GGR-18 Reticulated	4,018	992	2,309±332

metadata. For example, an animal that appeared to travel too far in a short period was marked as a potentially lousy ID with multiple animals and sent for additional review. For the GGR-16 and GGR-18 events, a global constraint of 10 km/h was applied to all sightings of Grévy's zebra and reticulated giraffe. This speed check identified a handful of annotations that were improperly merged into the same ID and were manually fixed by splitting the ID into two (or more) individual animals.

The animal IDs are then combined with their image's date/timestamps to determine when and where an animal was seen. Knowing the number of sightings on day 1 only, day 2 only, and resightings between both days allow a Lincoln-Petersen estimate to be calculated as in a traditional mark-recapture study (Table 6.2). In addition, embedded GPS meta-data – and knowing the camera and car a photograph originates from – can be used to analyze the spatial and temporal distributions of the data and the distributions by car and photographer. The result is a final list of the named animals with age and sex information, which can then be compared to previous years to develop a list of deaths, births, migration patterns, and other ecological insights.

6.1.2.5 Convergence & Sighting Distribution

Next, we turn our attention to determining how well the censusing events sampled the underlying animal population. Figure 6.17 plots the number of new animals identified vs. the number of processed photographs, ordered chronologically. Ideally, these curves should flatten over time, indicating that the rate of encountering unknown individuals in the database is slowing. The

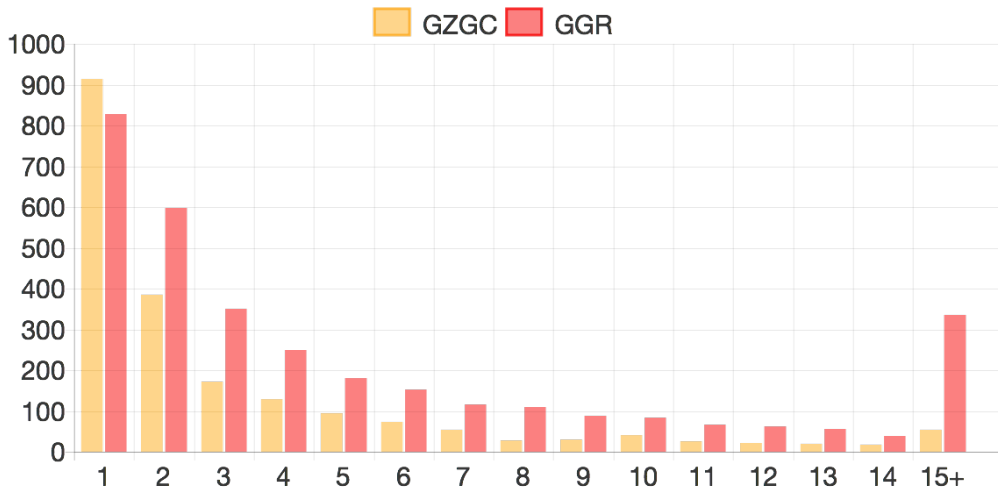


Figure 6.18: The number of photographers per animal ID for the GZGC and GGR-16 photographic censusing rallies. The total number of photos from the GGR is much higher than the GZGC, and the number of 15+ photos is much more saturated, indicating better coverage and that the number of resights should be much higher. ©2017 AAAI. Reprinted, with permission, from: J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

slope of the GZGC curve [2] flattens out over time but does not entirely converge. The final GZGC slope suggests that if there were additional photographs to analyze, then new individuals were likely going to be discovered (increasing the population estimate and narrowing the confidence interval). We can compare the GZGC curve to the GGR-16 and GGR-18 curves, which are flattening out much faster and starting to converge. The GGR curves suggest that collecting more photographs may not have substantially impacted the final population estimate because new IDs were becoming rare. This intuition is supported by Figure 6.5 (right), which explicitly shows a higher percentage of resights in the GGR compared to GZGC and indicates overall better coverage of the underlying population.

Figure 6.18 plots a histogram of the number of photographs per animal. It shows that most frequently, an animal was photographed only once during both rallies. However, the collection protocol encouraged volunteers to take three photographs of a sighted animal, which disagrees with this histogram. Ideally, the number of animals seen only once should be low, while the number of sightings distributed around three sightings should be abnormally elevated. Unfortunately, the actual distribution suggests that this collection request was challenging for the volunteers to

follow consistently. Encouragingly, the number of animals with single-sightings *decreased* between the GZGC and the GGR-16, even though the number of annotations more than tripled. This improvement suggests that more thorough sampling (i.e., more volunteers) and better training can help correct this bias.

6.1.3 Animal Population Estimates

At long last, the population estimates of the GGR-16 and GGR-18 on Grévy's zebra and reticulated giraffes can be computed using their respective animal ID databases. The population estimates from the GZGC [2] are also provided for Plains zebras and Masai giraffes as a comparison. In addition, Table 6.2 provides a summary of the number of annotations used for identification, the number of sighted individuals, and the Lincoln-Petersen index as the population estimate.

6.1.3.1 Results of GGR 2016 (GGR-16)

Over 40,000 images were collected for the GGR-16, which resulted in 16,866 annotations and 1,942 animal IDs after curation was performed. The animals sighted during the GGR-16 were sighted on either day 1 only, day 2 only, or were sighted on both days. A total of 1,416 individuals out of 1,942 were seen on day 1, a 72.91% sampling rate of the total population of sighted individuals. During day 2, a slightly lower number of animals was seen at a total of 1,338, representing a percentage of 68.90%. The number of animals that were resighted between the two days was 835. These statistics can be used to generate a Lincoln-Petersen population estimate, as shown in Table 6.2 with a confidence interval of 95%. The population estimate of 2,269$\pm$95 indicates that during the GGR-16, between 82% and 89% of the surveyed population as seen. These statistics suggest that the methodology from 2016 was mainly effective at sampling the resident populations of Grévy's zebra.

Unfortunately, upon analyzing this data as broken down by county (see Table 6.3), there were abnormally low population densities in the most northern censusing blocks. In addition, these blocks had a sparse set of citizen scientists across a vast geographical area, resulting in a substandard sampling. As a result, only 45 and 46 individuals were sighted for days 1 and 2 of the GGR-16 censusing rally, and only 26 were resighted. These small numbers were compared to an adjoining county that found roughly 240 individuals with over 170 resighted individuals each day. For the GGR-18, additional efforts were explicitly focused on the northern counting blocks to provide better coverage and correct the previous GGR event.

Table 6.3: The number of annotations, matched individuals, and the final mark-recapture population size estimates for Grévy's Zebra for the GGR-18, by county. The Lincoln-Petersen (L-P) estimates are calculated with a 95% confidence interval. The county breakdown has a slightly lower total due to images that did not properly localize within the exact county areas yet was matched by the identification pipeline.

Kenyan County	GGR-16 L-P Estimate	GGR-18 L-P Estimate
Isiolo	299±54	597±182
Laikipia	1199±59	1328±105
Marsabit	66±17	250±166
Meru	344±30	384±36
Samburu	452±80	642±172
Northern Blocks	80±21	545±265

6.1.3.2 Results of GGR 2018 (GGR-18)

The number of participating photographers increased by 33% during the GGR-18 censusing event. The photographers contributed over 49,000 images, representing a 21% increase compared to the GGR-16 event. The increased number of collected photographs was due to adding more photographers and including reticulated giraffes as a second species of interest. More than 23,000 images contained the primary species of Grévy's zebra, whereas over 18,000 images contained sightings of reticulated giraffes. These images resulted in 10,044 annotations that were used for identification for Grévy's and 4,018 for giraffes. The drop in identifiable annotations between the GGR-16 and GGR-18 for Grévy's zebra can be explained by more a restrictive filtering process within the detection pipeline. This filtering drastically reduced the overall number of human reviews required by the identification pipeline but slightly increased the population estimate error bounds compared to the GGR-16. Within those annotations, a total of 1,972 unique Grévy's zebra and 992 unique reticulated giraffes were sighted.

A total of 1,251 individuals out of 1,972 zebras were seen on day 1, a 63% sampling rate of the total population of sighted individuals. During day 2, a slightly higher number of zebras was seen at 1,299, representing a percentage of 65.9%. It is worth noting that these sampling numbers are relatively consistent with the GGR-16 sampling. For the GGR-18, the number of zebras that were resighted between the two days was 578. This value is much lower, suggesting that either the population has gotten significantly larger (unlikely), the sampling procedure was not thorough enough (a constant issue), or the detection pipeline was slightly too restrictive.

The population estimate of 2,812±171 for zebras indicates that, during the GGR-18, somewhere between 66% and 75% of the surveyed population was seen. Again, these numbers are lower than the GGR-16, possibly indicating a step backward in sampling or the quality of images. These statistics suggest that our methodology from 2018 was still effective at sampling the resident populations of Grévy's zebra, but not as thorough as the 2016 census. However, the analysis is not entirely bad news when looking at county breakdowns of the GGR-16 and GGR-18 census estimates, as shown in Figure 6.3. The northern blocks were sampled much more heavily during the GGR-18, and an estimated 400 animals were recovered due to better sampling. The county breakdowns of southern counties with the largest populations (Laikipia, Samburu, and Meru) as their population estimates are stable, providing a measure of confidence in the methods used. The error bounds for these counties are also overlapping, suggesting that any bias in the population estimate is not statistically significant for the vast majority of the animals.

For giraffes, on day 1, 613 individuals were sighted, 516 on day 2, and 137 were resighted across both days. These numbers are much smaller, and the sampling ratios are much smaller. The drastically smaller resighting value compared to the total number of individuals is reflected by a high estimate (and error bound) for the giraffe population. The statistics suggest that the number of giraffes in the surveyed population area is 2,309±332. The giraffe results are to be treated as tentative, pending a re-censusing in 2020.

6.2 Culminating Experiment on GGR 2018

This section sets aside the narrative tone from the above discussion on the GGR and how its results were calculated. Instead, the purpose of this culminating experiment is to provide a condensed walk-through of the most current, recommended process for animal photographic censusing in 2021. As such, we can opt to rely on the pre-trained detection pipeline and advanced Census Annotation machine learning models that have been introduced. What follows is an end-to-end analysis by following the steps listed in Figure 6.2 (starting just after the "Image Aggregation" step) on a large animal population.

The population estimates from the GGR-16 and GGR-18 were a product of their respective times and the states of the computer vision algorithms available to them. The field of computer vision has advanced drastically between then and now, with some of the components introduced in this thesis not even being available at the time of their original analysis. However, to demonstrate the effectiveness of concepts like Census Annotations and the LCA curation algorithm, we wish to

audit a GGR censusing event with modern tools. The benefit of having photographic ID evidence for a population is that newer machine learning approaches can be evaluated for improvements in accuracy and human involvement. This section performs a new, standalone analysis of the data collected by the GGR-18 photographic censusing event (only for Grévy's zebra) and compares the population estimates to the values reported in Section 6.1.3.2. This analysis focuses on the GGR-18 over the GGR-16 event because it is the most representative geographic sampling (covering the northern blocks) and is the active population estimate provided by the Kenyan government.

We processed all of the raw GGR-18 imagery fresh for an end-to-end run-through of the entire procedure. We imported a total of 56,588 valid images and ran the automated detector pipeline on the images. The localizer was configured using the pre-trained GGR-18 Grévy's zebra model and used an NMS threshold of 40% and an operating point of 40%. The localization model produced 104,858 annotations, with 73,356 of those labeled with the generic `zebra` label. Using a single NVIDIA Titan RTX and a 20-core server, the bounding box regression computation took approximately 6.5 hours.

The GGR-18 census event asked volunteers to take right-side shots of Grévy's zebra, so we need to use the labeler network to filter out other species (including plains zebra) and any viewpoints that did not show the right side. The labeler was trained using a collection of Grévy's and Plains zebra data and various viewpoints. The `zebra_v1` model (an ensemble of DenseNet-style neural networks) was configured such that the most confident `species:viewpoint` prediction that was returned was used as the final label. The labeler computed results on the same accelerator hardware in approximately 5.5 hours. The labeler produced a classification of `zebra_grevys` for 67,409 annotations, but approximately half of those showed the incorrect viewpoint. The algorithm predicted 23,458 "right", 8,735 "front-right", and 8,273 "back-right" viewpoints labels; all annotations that were not Grévy's zebra or that did not show the right side were discarded, leaving 37,199 for further processing. These annotations were sourced from 20,015 original images.

Next, the `V4` CA classifier (see Section 5.2) was run on these images to identify the most likely Census Annotations for ID. The inference ran for 1 hour and 40 minutes and produced a set of 15,072 Census Annotations (threshold = 0.31). For reference, using a CA classifier threshold of 0.1 results in 16,848 annotations while using 0.9 ends with 11,381. Thus, using the recommended threshold of 0.31 keeps the majority of the borderline CAs. Next, each CA was passed to the CA Region regression network, which took 42 minutes to compute. Once the bounding boxes were generated, we performed NMS (IoU = 1.0) to prevent overlapping CA Regions by suppressing the

lower-scoring box as determined by the CA classifier. The purpose of using an IoU of 100% was to guarantee zero overlap between annotations within an image, drastically reducing the incidence rate of mother-foal photobombs. This filtering resulted in 14,742 Census Annotation Regions for right-side Grévy's zebras; these annotations were sourced from 12,772 images, indicating a utility rate of the citizen science capture of 22.6%.

Next, we synchronized all of the GPS locations and timestamps for all of the images (and source cameras) for these Census Annotation Regions. All annotations taken outside the GGR-18 date range (Day 1: 1/27/2018 and Day 2: 1/28/2018) were discarded. The standard GPS synchronization step was delayed until now to focus on only the images that contributed useful Census Annotation Regions. There were 7,737 annotations taken on Day 1 and 6,116 on Day 2, coming from 11,991 images. Additional filtering eliminated poor qualities based on pixel size, aspect ratio, and gradient magnitudes (blurriness). While the CA classifier does an excellent job filtering out incomparable sightings, it still makes some mistakes. Most of the CA classifier's mistakes come from blurry annotations, and – while they are subjectively challenging pairs and take humans a long time to decide – they are comparable. The average aspect ratio (height / width) was calculated and any box outside of 2.0 standard deviations was discarded (min = 0.328, max = 1.449, mean = 0.612, std = 0.109), removing 698 annotations. A similar filter was applied to the total number of width and height pixels for each annotation. An additional 281 annotations were removed (minimum width 245 pixels, minimum height 161 pixels) using a minimum threshold of the mean minus 1.5 standard deviations. Lastly, we computed the average gradient magnitude across the image using an x-axis and y-axis Sobel filter (kernel size 3). The average of the gradient magnitude mean for all annotations was calculated as 86.0 on `uint8` RGB 3-channel cropped chip with a maximum-linear dimension of 700 pixels (min = 12.8, max = 182.5, mean = 86.0, std = 27.9). All annotations with a mean gradient less than 1.5 standard deviations under the average mean gradient (less than 44.2) were removed. This left 11,916 annotations (Day 1: 6,677, Day 2: 5,239) and 10,558 images. All other annotations and images that did not contain a Census Annotation Region or passed these photometric quality filters were discarded and not used for further analysis. All processing up to this point has been completely automated.

Next, the LCA algorithm was initialized by loading a pre-trained weighter function. The weighter was created from 500 positive and 500 negative pair decisions from the GZCD dataset (see Section 4.4). Next, the match candidates were sampled using a HotSpotter rank list for all of the CA Regions (tuned for $K = 5$, $K_{\text{norm}} = 5$, $n_{\text{top}} = 10$, spatial verification was ON, scoring method

csum) that found 67,247 pairs. The VAMP model trained on Grévy's zebra CA Regions was used as the verification algorithm and applied to all matching pairs. In total, 13,848 negative weights, 10 neutral weights, and 53,389 positive weights were found by the VAMP verifier for LCA. The LCA algorithm then proceeded to try alternative pairs of clusters, asking for VAMP and human decisions, as it worked. The pairs that the algorithm wanted a human decision were given to a web interface and reviewed by the author.

In total, it took just under 12 hours for the LCA algorithm, working with a single human reviewer to converge. The LCA process attempted 23,783 alternative clusterings and requested an additional 19,160 VAMP decisions during its automated processing. The ID curation process required just 1,297 human decisions before converging. For reference, the original GGR-18 analysis using the Graph ID algorithm required 18,556 human decisions. The resulting ID database was then checked to ensure no erroneous IDs with poor singletons needed to be excluded (i.e., the detection pipeline failed to filter them out). In total, 15 IDs were excluded for having too poor quality, leaving 2,022 unique IDs in the database. The demographics labeling step was skipped, and the gender checks were not performed for the sake of a more straightforward verification, but the speed check resulted in zero IDs being marked for review. A total of 1,338 individuals were seen on day 1, 1,326 animals were seen on day 2, and 642 animals were seen on both days. The final Lincoln-Petersen estimate using this new ID database was 2,764$\pm$154, consistent within 1.7% with the reported estimate on GGR-18 (2,812$\pm$171). Furthermore, this result was 93% less human effort than the GGR-18 processing and was completed with the effort of a single working day for one person.

Finally, we need to incorporate the estimated machine learning loss terms from Equation (4.21) (in Chapter 4) into the final population estimate. Recall that the equation has been modified to accept three new terms $\hat{p}_{mm}(\theta)$, $\hat{p}_{ms}(\theta)$, and $\hat{p}_{dm}(\theta)$. The term $\hat{p}_{ds}(\theta)$ is assumed to be zero as each of the final singletons were manually checked for quality. Furthermore, using Census Annotation Regions helped to reduce the rate of photobombs and scenery matches drastically. The LCA algorithm was also configured to do an additional brute-force check for potential incidental matches, ensuring that each name cluster had more stability with extra automated checks beyond what was requested by the ranking algorithm. The verification allows us to make the assumption that $\hat{p}_{ms}(\theta)$ is also close to zero, leaving $\hat{p}_{mm}(\theta)$ and $\hat{p}_{dm}(\theta)$ as the primary terms to impact the result. The reported GGR detection recall performance on AoI Grévy's zebra is 96%, indicating that 4% of the annotations are missed. Furthermore, the CA classifier has a false-negative rate of 1.8% on the GZGC dataset, so a conservative detection miss rate of 6% is estimated for $\hat{p}_{dm}(\theta)$. For $\hat{p}_{mm}(\theta)$, the top-10 recall

rate for HotSpotter on Grévy's zebra CA-Rs in the GZCD is 99.2%. We must also consider the VAMP failure rate for CA-R `match` decisions of 1.4%. These effects combined, but with the human verification of borderline matches, give us a conservative estimate for $\hat{p}_{mm}(\theta)$ at 2%. Using these values suggests a correction value of +56 IDs for the population estimate and a widening of $\pm$13 for the confidence interval. The final corrected population estimate with high degrees of automation is therefore 2,820$\pm$167 (0.3% off), compared to the originally reported estimate for GGR-18 of 2,812$\pm$171. Furthermore, the original GGR-18 analysis marshaled dozens of participants and took around three months to complete. In sharp contrast, the new analysis took approximately two days – with the majority of that time dedicated to hands-off, automated computing – and only relied on one person.

6.3 Summary

The Great Grévy's Rally (GGR) is the most extensive photographic census of the Grévy's zebra species ever performed in Kenya. The country of Kenya is the primary residence of Grévy's zebra, indicating that the population estimates from the GGR are also comprehensive for this critically endangered species. Furthermore, the data collection rallies in 2016 and 2018 are a real-world demonstration of the principles of citizen science and the benefits of using volunteer photographers in data collection. Across the GZGC and GGR censusing rallies, over 100,000 photographs were processed and collected by more than 400 volunteer citizen scientists (90,000+ images and 350+ volunteers for GGR alone). Furthermore, the GGR-16 and GGR-18 rally procedures were significantly improved by increasing the automation of the detection and identification processing, streamlining data collection with GPS-enabled cameras, and proving that the original methodology from the GZGC scales to thousands of animals. Unfortunately, even though the data analysis for GGR-18 was done with automated tools, it still required large amounts of work (nearly 20,000 human decisions), cost USD $50,000+, and took over three months.

Luckily, new advances to the ID curation process since the conclusion of GGR-18 have been developed: Census Annotations, Census Annotation Regions, the LCA curation algorithm, and a new Lincoln-Petersen index estimator. The primary motivation of these components is to increase the automation of photographic censusing further and reduce the known issues (e.g., incidental matching) that were encountered during the GGR-16 and GGR-18 analysis. As a result, these new methodologies were used to reprocess the original GGR-18 collection from scratch. In total, 56,588 images were automatically processed by the pre-trained detection pipeline, and 11,916

annotations were found for comparable, right-side Grévy's zebra. The ID curation process required 1,297 human decisions before converging and estimated 2,764±154 Grévy's zebra in the population. After modifying for various known ML errors, the estimate was updated to 2,820±167. This result is consistent (within 0.3%) with previous estimates on GGR 2018 data and was achieved with a 93% reduction in human effort. This new automated image analysis procedure will be employed on the newly collected data for GGR in 2020 (GGR-20) and beyond.

In summary, the GGR-18 censusing rally produced a population estimate of 2,812±171 for Grévy's zebra, indicating approximately 70% of the population has been included in the ID census database. Furthermore, the 2018 population of reticulated giraffes in Kenya is estimated to be 2,309±332. These estimates are consistent with previous sampling counts and provide a new image-based ID database for historical trends.

6.3.1 Lessons Learned

Having completed three successful events, having developed both logistic support methods for running the events, and having new computer vision/machine learning algorithms for analyzing the resulting image data, we conclude this chapter with a discussion of lessons learned about applying photographic censusing in the real world. For example, the prototype photographic censusing event, the Great Zebra & Giraffe Count (GZGC) [2], was marred by metadata synchronization issues (no GPS-enabled cameras), an inability to eliminate problematic annotations (no CA for incidental matching), relied exclusively on human decision making for detections (no pre-trained models) and pairwise review (no VAMP), had no concept of ID curation or consistency checks (no Graph ID or LCA), did not know beforehand how helpful citizen scientists were going to be at image collection (no established baseline), and took three months of hand-crafted analysis (no existing tooling or software). These problems translate directly into a high logistical burden on conservation administrators and, if left unaddressed, would undercut photographic censusing as an attractive alternative to more invasive monitoring methods.

Since our initial start in 2015, however, the methodology for photographic censusing has dramatically improved. The lessons that have been learned have translated directly into a proven, end-to-end system that the Kenyan government is actively using to track animal populations. The challenges encountered during the GZGC, the GGR-16, and the GGR-18 – emphasizing that some were significant barriers – offered a guiding framework for designing a more robust, reliable, verifiable, automated, sustainable, and repeatable methodology. With the conclusion of this chapter,

let us review some of the most influential takeaways and high-level recommendations from this concerted and sustained research effort:

- **Citizen Science** - the completed photographic censusing events have demonstrated that distributing data collection for specific ecological research is highly effective. Furthermore, incorporating volunteers is a natural way to engage with a local community with science projects. When required, any training procedures should focus on being easy-to-understand and, ideally, should fit onto a single page. The presented research has shown that citizen scientists can quickly learn and conform to specific and focused data collection goals.

- **On-the-Ground Participation** - there is little substitute when performing a photographic census for on-the-ground participation of the principal investigators. The nuances of distributed data collection are hard to predict and may significantly impact the effectiveness of the overall event. Furthermore, participating in the photographic census in-person allows for engagements with conservancy managers, park rangers, and field ecologists, who may be mandated by law to maintain accurate population estimates. Establishing relationships with these science brokers is crucial because it 1) allows for an accurate assessment of the correct geographical coverage area to capture the known range for the species of interest and 2) provides a touch-point for setting up a routine and secure exchange of the latest ecological data as it is collected. Lastly, engaging with local data brokers and policymakers helps prevent claims of "exporting" the data from the conservation area, depriving a local entity of their sense of agency and self-determination in conservation action. Furthermore, establishing productive partnerships with ecologists is an ethical and sustainable way to collect animal data for ML research.

- **Local Infrastructure** - it is essential to consider and be aware of the local infrastructure (e.g., power, internet, cell service) and its limitations during a photographic censusing event. Dispatching photographers into remote areas may have inherent safety concerns, and the final aggregation of the collected data should be based on a reliable mechanism. For example, having intermittent access to stable power may prioritize battery-powered laptops for the event administrators. Likewise, ecology research for endangered species can be performed in areas without reliable or fast access to the Internet. This limitation means that communication with cloud-based servers may not be available, and the ML processing must be delayed or done locally with power-hungry, high-performance GPUs.

- **Ease of Participation** - it should be easy for a volunteer to participate, with minimal requirements on the camera hardware. The only significant restriction for proper synchronization is having at least one GPS-enabled camera within a party of photographers. This requirement could be satisfied with a smartphone and, ideally, precludes the need for specialized camera trap hardware, access to planes, or other complications like veterinary licensing. In addition, a detailed aggregation and synchronization plan should be established prior to the data collection event. Furthermore, this plan should include details on properly cleaning, sanitizing, and discarding inappropriate images that may be accidentally contributed. Lastly, using registration cards allows for large numbers of participants to contribute data with minimal record-keeping and the need for real-time coordination.

- **Ease of Scale** - a photographic census is designed to have a fixed geographic area sampled on two consecutive days. The participation of photographers should be structured such that minimal to no coordination is required between photographers. For example, large sampling areas can be broken up into zones to ensure uniform coverage. Increasing the sampling density for a given area simply requires assigning more participants. After the two days of the event, all of the image data will need to be physically collected at a centralized location or otherwise submitted to the administrators for aggregation.

- **Open Populations** - photographic censusing is designed for open animal populations where the actual number of individuals is not known. While censusing open populations means a formal validation of the results is not possible, the methodology has produced results consistent with historical estimates for multiple species in Kenya. Furthermore, no inherent limitation would make photographic censusing incompatible with closed populations for more precise evaluations.

- **Data Wrangling** - after collecting the image data from photographers, there has always been a non-trivial step of cleaning, organizing, and otherwise preparing the raw data for machine learning processing. This process is typically done manually (e.g., copying images off an SD card into a folder) and is subject to human errors and poor standardization. The aggregated imagery might also require large amounts of storage, and the logistics of transferring and backing up the entire censusing event should be considered carefully. Another challenge is when photographers forget to take their synchronization image, requiring a manual process to establish the internal time for that participant's camera. In some instances, this synchronization

can only occur after the animal IDs have been fully assigned. A camera's time offset can be approximated by triangulating the speed consistency checks across multiple animals.

- **Bootstrapable Machine Learning Components** - the machine learning components that are used throughout the process are meant to be bootstrapable (i.e., trainable from scratch without access to an appropriate pre-trained model). Thus, all detection pipeline components can be trained with relatively small labor (estimated to be approximately 1,000-2,000 hand-annotated images) and do not require a fully segmented ground-truth. In practice, a small team (1-5 people) of reviewers can gather all of the required detection ground-truth data in about a day. Furthermore, it is recommended to implement web-based tools for multiple workers to contribute ground-truth data simultaneously.

- **Animal Detection** - the detection pipeline components, Census Annotations, and Census Annotation Regions all function as a way to automate the conversion of raw collected imagery to relevant annotations. The task of "animal detection" as a high-level process functions as an aggressive filter and only allows easily identifiable and comparable annotations to be considered by the ID curation process. The proposed detection components in this dissertation are designed to be modular and standalone, allowing them to be replaced without needing to modify other components in the pipeline. These components do not need to achieve state-of-the-art performance on their respective tasks because their primary goal is to reduce work. For example, the task-specific goal for bounding box regression is to reduce the L2 regression error, but there is an overriding goal to decrease the incidence rate of incidental matching. Furthermore, we do not need the best and most up-to-date detectors computer vision offers to calculate an accurate population estimate. Lastly, not all detections are equally important (i.e., missing a blurry background animal is not an error), and the detection components should be optimized for the specific purpose of filtering out incompatible, unidentifiable, incomparable, or otherwise problematic annotations.

- **Human Verification** - a critical determination for a new candidate species is if a human can accurately and timely tell if two annotations show the same individual or not. For example, performing a photographic census on the common American red squirrel is likely not very productive because people have a hard time telling two squirrels apart. As such, the utility and accuracy of any photographic census are explicitly tied to how accurately a human can correctly verify potential matched pairs.

- **Continual Curation** - selecting which curation algorithm to use is vital, as its implementation details can significantly influence automation. For example, the Graph ID algorithm [13] was found to be too rigid, too iterative, and too quick to explicitly enforce consistency, leading to the need for many thousands of additional reviews. Alternatively, the LCA algorithm is more appropriate for photographic censusing because it allows for qualitative decisions that do not block the processing workflow, which dramatically reduces the need for human effort while maintaining accuracy. Furthermore, a photographic census produces a consistent and highly-reliable database of animal IDs but may start from different underlying database states. For example, a photographic census may be started with an empty database, a sizeable pre-existing database that has already been thoroughly curated, or a pre-existing database with many unresolved issues (i.e., merges, splits, consistency checks). The ID curation algorithm must be capable of handling these different starting conditions.

- **Bootstrapable ID Databases** - large-scale animal ID databases are rare, and photographic censusing is an end-to-end process that facilitates the creation of large, highly-consistent databases for animal ID research. Furthermore, some of the newest and most accurate machine learning approaches for re-identification (i.e., triplet loss) need to be trained on an existing database of IDs. This limitation means that approaches that do not rely on deep learning (e.g., HotSpotter [261], VAMP [13], CurvRank [262], and even Graph ID [13]) are required to enable starting an ID database from scratch. Once a critical mass of IDs has been gathered, these tools can be replaced for more advanced and accurate approaches like PIE [263]. Lastly, from experience, it is important not to underestimate the time and challenge it will take to collect and curate reliable animal ID data for computer vision research.

- **Reporting ID Ranking Performance** - there is often a disconnect with how animal ID ranking performance is standardized and reported. For example, it is crucial to consider the underlying distribution for the number of annotations per name, the percentage of singletons in the database, and the time between sightings. As a general rule of thumb, it is recommended that ID ranking and recall performance be reported as the average percentage for all annotations in a database for top-1, top-5, top-k. Any experimental evaluation should start by defining a fixed set of annotations where each animal ID has a minimum of 2 annotations, a maximum of 5 annotations, and a minimum of 24 hours between any pair of its annotations. This structure guarantees that 1) there is always a correct answer to be found in the top-1 rank, 2) that an

over-sampled name does not improperly skew the reported recall performance, and 3) that trivial matches separated by seconds or minutes are removed. Furthermore, the performance should be reported as the average across all annotations (i.e., annotation recall) and all names (i.e., name recall).

- **Ownership & Security of Animal IDs** - it is essential to consider the ethics of claiming ownership of an animal's identification. There is an inherent obligation to protect an endangered species when performing an ecology study. It includes being responsible for how sensitive metadata is accessed and requires a willingness to embrace collaboration opportunities with other researchers. For example, it would be unethical for a researcher to claim exclusive ownership of an animal's ID to the detriment of more effective conservation action and insight. Stated simply: a wild animal's ID does not belong to you, it belongs to the animal that is facing extinction. Likewise, careful consideration for sensitive metadata (e.g., GPS locations, timestamps) should be paid for poached and critically endangered species, as a comprehensive ID database is a phenomenal surveillance tool for bad actors. A photographic census administrator's responsibility is to adequately safeguard any sensitive information to protect the life, health, and unique identity of any endangered animal.

- **Extensibility & Sustainability** - the real-world reality of photographic censusing is that it is currently a highly-specialized endeavor, dependent on advanced computer science and computer vision expertise. Therefore, it is imperative to consider hosting all code under a permissive open-source license and with a freely available repository service. In addition, the publication of free pre-trained models offers an opportunity for collaboration and reproducing results. Lastly, the financial reality of machine learning-powered ecology is currently dependent on support from governments, NGOs, not-for-profits, private industry grants, and philanthropic foundations. For example, this dissertation focuses heavily on just a single species (Grévy's zebra) because it takes so much time, energy, and money to collect and curate good ID datasets like the GZCD. Therefore, the lack of a mature, self-sustaining, open-source community for automated wildlife monitoring represents a substantial existential crisis for the long-term adoption of photographic censusing.

In conclusion, this dissertation proposes a new paradigm for animal population monitoring; the high level of accuracy and automation that has been demonstrated will ideally transform ecology into a data-driven science.

CHAPTER 7
CONCLUSION

The proposed photographic censusing methodology encompasses the entire process from start to finish: from the engagement of citizen scientists for decentralized image collection; to the parallel annotation of new training data; to the training and inference of automated decision making with computer vision algorithms; to the final population estimates with their valuable individual ecological, social, and temporal data. Furthermore, this dissertation demonstrates the effectiveness of a detection pipeline for filtering raw images into identifiable annotations and reducing errors by mitigating common identification errors. Any errors that are made by the automated algorithms can also be factored into the final population estimate. Thus, the power of photographic censusing is driven by comprehensive detection and identification computer vision pipelines and a thorough understanding of how modern ecological studies are performed in the field.

7.1 Contributions

The research presented in this dissertation is heavily applied and represents the culmination of over a decade's worth[30] of computer vision and ecology research. Furthermore, the trajectory of this work has been heavily influenced by the real-world implications and implementation details of performing photographic census events on endangered animal populations.

1. **Animal Detection Pipeline** - a comprehensive detection pipeline for animals for use in photographic censusing. The pipeline is designed to be easily bootstrapable for new species with relatively minimal annotation work. The output of the detection pipeline is customized for the task of animal instance recognition and is comprised of the following modularized components:

    (a) Whole Image Classification (WIC) - A CNN that performs a multi-label classification problem for high-level filtering

    (b) Localization - A CNN that performs bounding box localization and classification to find animals

    (c) Annotation Classification (Labeler) - A CNN that performs a single-label classification problem for annotating species and viewpoint

[30] and has required the focus of more than one Ph.D. dissertation, see [13].

(d) Coarse Background Segmentation - A FCNN that attempts to provide an approximate segmentation for a given species to mask out background pixels.

(e) Annotation of Interest - We present the concept of AoI and evaluate its effectiveness for filtering irrelevant annotations in an identification pipeline.

(f) Specialized Needs - Additional detection components to rotate annotations or find animals from overhead imagery are useful for specific needs.

2. **Census Annotation** - a novel concept that is designed to reduce incomparable and incidental matching during animal identification.

   (a) Census Annotation (CA) - selects annotations that are identifiable and show a consistent part of the animal body, reducing the amount of human work that is needed during a photographic census.

   (b) Census Annotation Region (CA-R) - reduces the impact of incidental matching by creating more focused regions within existing detected Census Annotations, drastically reducing the amount of human effort by increasing the separability of automated ID verifiers

3. **Photographic Censusing** - a comprehensive process for building an animal ID database from scratch, relying on the concepts of verification and the continual curation of IDs. The formal definition also includes a new Automated Lincoln-Petersen Estimator to better estimate populations when machine learning methods are involved.

4. **Photographic Censusing Rallies** - an organized data collection event where "citizen scientist" volunteer photographers are trained and tasked to take photos with GPS-enabled cameras over two back-to-back days. The results of the Great Grévy's Rally 2016 (GGR-16) and Great Grévy's Rally 2018 (GGR-18) censusing rallies are significant contributions of this work.

5. **Animal Datasets** - new public datasets for animal detection and ID research. Common public datasets for computer vision tasks like object detection generally do not provide associated ID information when they include boxes of animals. Likewise, animal ID datasets often only include pre-cropped images of animals and rarely focus on herding species.

   (a) WILD - a dataset with six species containing comprehensive bounding boxes and AoI flags on challenging scenarios.

(b) DETECT - a Plains and Grévy's zebra only database focusing on even more visual nuances for detection.

(c) Grévy's Zebra Census Dataset (GZCD) - a dataset of Grévy's zebra that focuses on the problem of incidental matching.

(d) Great Grévy's Rally 2016 (GGR-16) - A first-of-its-kind photographic census of the Grévy's zebra in Kenya, producing a baseline population estimate.

(e) Great Grévy's Rally 2018 (GGR-18) - A second census of the Grévy's zebra and reticulated giraffe in Kenya, providing a population estimate of Grévy's zebra with improved sampling and measuring the increase of the population.

7.2 Future Work

The field of automated wildlife conservation is in its infancy, and there seems to be a lack of widely available animal ID datasets. Building large-scale datasets on animals with the first principle of ID seems to be the fastest way to unlock the interest within the larger research community on animal detection and re-ID. Furthermore, my hope is that the analysis provided on the Grévy's zebra species has demonstrated the overwhelming benefits of photographic censusing as a population monitoring methodology, where the principles that have been described weave a general framework that can be easily adapted for other endangered species. If both cases are true, then the next step is – quite simply – to get to work protecting some animals that need our help to survive.

The study of endangered species is tricky when state-of-the-art research methods also produce fantastic tools for poachers. The availability of an ID database for conservation policy has the apparent downside of being a clearinghouse for the size and location of a given population. Continued research should be focused on one-shot learning to reduce the exposure of a species to only what is essential for machine learning training. A focus on one-shot or few-shot learning also comes with the obvious benefit of not building animal population monitoring systems that are brittle to a low distribution of sightings.

A major missing component of this dissertation is a robust analysis of true segmentation methods. The ability to segment a mother and foal is likely the only way that problem will be solved long-term. Likewise, some species are poor candidates for bounding boxes (e.g., giraffes) because they fill the annotations with a lot of background noise. The downside is that segmentation algorithms have historically been very data-hungry to get good performance, but the future looks

bright for more accurate filtering methods prior to ID.

The definition of a given species' Census Annotation Region relies primarily on human intuition and depends to some extent on the ranking and verification algorithms used during ID curation. There needs to be a more principled way to locate the visual information needed for effective automated ID curation. Related work on attention mechanisms with deep convolutional neural networks may provide a mechanism for automatically defining these regions on a per species basis. Furthermore, there may be a need to decouple the results of an ID algorithm with the visualization of its suggested correspondences. For example, the PIE algorithm does not natively visualize the matching regions between two annotations that are found to be close in its learned embedding space. While a given ID algorithm may offer useful intermediate primitives to visualize, this cannot be guaranteed, and a more standardized visualization approach may be possible.

The process of photographic censusing is presented here as a comprehensive, bootstrappable, and end-to-end option for wildlife conservation managers. The motivating use case for this dissertation has been the management of large *megafauna* populations in Kenya. As such, one of the core implementation decisions with photographic censusing rallies is that it relies on two days of collection. This two-day structure attempts to follow the guidance of historical surveys done in that country, but it is not the only valid time frame option for a census. There exists a clear need for more flexible collection options because some species cannot be comprehensively censused in a single day. For example, whale watching seasons typically involve several months of image collection and cannot be expected to cover the entire migratory range in only a handful of days. Instead, photographic censusing needs to be extended to support a longer, more continual collection of images. The Petersen-Lincoln estimator is not compatible with such a design, so additional statistical methods and validating experiments are needed to apply this work more broadly.

Finally, the research studies that have successfully used citizen science efforts have almost exclusively been focused on species classification and do not meaningfully engage with ID verification. For example, there does not seem to exist experimental data on how well the general public can verify if two zebra are the same in a fixed time, or two beluga whales seen from above, or the flippers of two sea turtle sightings. Furthermore, there is no established international standard for how photographic censusing should be performed, focusing on less invasive collection and automated analysis. A recognized portfolio of species that are good candidates for photographic re-ID and recommendations for how best an average citizen may safely collect and contribute valuable data would unlock new avenues for data collection.

REFERENCES

[1] D. Rubenstein, B. Low Mackey, Z. Davidson, F. Kebebe, and S. King, "Equus grevyi," *IUCN Red List Threatened Species 2016*, vol. e.T7950A89624491, no. 1, pp. 1–15, Aug. 2016.

[2] J. R. Parham, "Photographic censusing of zebra and giraffe in the Nairobi National Park," M.S. Thesis, Dept. Comput. Sci., Rensselaer Polytechnic Inst., Troy, NY, USA, 2015.

[3] C. D. FitzGibbon and J. Lazarus, "Antipredator behavior of Serengeti ungulates: Individual differences and population consequences," in *Serengeti II: Dynamics, Management, and Conservation of an Ecosystem*, vol. 2, C. D. FitzGibbon and J. Lazarus, Eds. Chicago, IL, USA: Univ. of Chicago, 1995, ch. 13, pp. 274–296.

[4] J. R. Ginsberg, "Social organization and mating strategies of an arid adapted equid: The Grevy's zebra," Ph.D. Dissertation, Dept. Ecol. Evol. Biol., Princeton Univ., Princeton, NJ, USA, 1988.

[5] K. J. Tombak, "The behavioral ecology and host-parasite dynamics of the zebras of east Africa," Ph.D. Dissertation, Dept. Ecol., Princeton Univ., Princeton, NJ, USA, 2019.

[6] F. Keesing, "Impacts of ungulates on the demography and diversity of small mammals in central Kenya," *Oecologia*, vol. 116, no. 3, pp. 381–389, Sep. 1998.

[7] S. M. Kivai, "Feeding ecologicalgy and diurnal activity pattern of the Grevy's zebra (Equus grevyi, Oustalet, 1882) in Samburu community lands, Kenya," M.S. Thesis, Dept. Biol., Addis Ababa Univ., Addis Ababa, Ethiopia, 2006.

[8] D. Rubenstein, "The ecology of female social behavior in horses, zebras, and asses," *Physiol. Ecol. Jpn.*, vol. 29, no. 1–2, pp. 13–28, Jan. 1994.

[9] V. H. Zero, S. R. Sundaresan, T. G. O'Brien, and M. F. Kinnaird, "Monitoring an endangered Savannah ungulate, Grevy's zebra (Equus grevyi): Choosing a method for estimating population densities," *Oryx*, vol. 47, no. 3, pp. 410–419, Jul. 2013.

[10] D. S. Robson and H. A. Regier, "Sample size in petersen mark–recapture experiments," *Trans. Amer. Fisheries Soc.*, vol. 93, no. 3, pp. 215–226, Jul. 1964.

[11] R. Pradel, "Utilization of capture-mark-recapture for the study of recruitment and population growth rate," *Biometrics*, vol. 52, no. 2, pp. 703–709, Jun. 1996.

[12] G. A. F. Seber, *The Estimation of Animal Abundance and Related Parameters*. New York, NY, USA: Macmillan, 1982.

[13] J. P. Crall, "Identifying individual animals using ranking, verification, and connectivity," Ph.D. Dissertation, Dept. Comput. Sci., Rensselaer Polytechnic Inst., Troy, NY, USA, 2017.

[14] C. Poultney, S. Chopra, Y. L. Cun *et al.*, "Efficient learning of sparse representations with an energy-based model," in *Adv. Neural Inform. Process. Syst.*, Vancouver, British Columbia, Canada, Dec. 2006, pp. 1137–1144.

[15] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[16] B. M. Marlin, K. Swersky, B. Chen, and N. D. Freitas, "Inductive principles for restricted Boltzmann machine learning," in *Int. Conf. Artif. Intell. and Stat.*, Sardinia, Italy, May 2010, pp. 509–516.

[17] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *Int. Conf. Artif. Intell. and Stat.*, Clearwater Beach, FL, USA, Apr. 2009, pp. 448–455.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inform. Process. Syst.*, Lake Tahoe, CA, USA, Dec. 2012, pp. 1097–1105.

[19] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Oct. 2013.

[20] P. Sermanet *et al.*, "Overfeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[22] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.

[23] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 2147–2154.

[24] R. Girshick, "Fast R-CNN," in *IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[25] C. Szegedy *et al.*, "Going deeper with convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inform. Process. Syst.*, Montreal, Quebec, Canada, Dec. 2015, pp. 91–99.

[30] J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio, "Quadratic polynomials learn better image features," Département d'Informatique Et De Recherche Opérationnelle, Université De Montréal, Montreal, Quebec, Canada, Tech. Rep. 1337, 2009.

[31] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 806–813.

[32] Y. Lecun *et al.*, "Comparison of learning algorithms for handwritten digit recognition," *Int. Conf. Artif. Neural Netw., Paris*, vol. 60, no. 1, pp. 53–60, Oct. 1995.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[34] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *IEEE Int. Conf. Doc. Anal. Recog.*, vol. 3, Edinburgh, Scotland, Aug. 2003, pp. 958–958.

[35] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[36] M. Bertozzi *et al.*, "A pedestrian detector using histograms of oriented gradients and a support vector machine classifier," in *IEEE Intell. Transp. Syst. Conf.*, vol. 27, Bellevue, WA, USA, Sep. 2007, pp. 143–148.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[38] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2, New York, NY, USA, Jun. 2006, pp. 1491–1498.

[39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[40] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[41] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.

[42] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Ann. Int. Symp. Comp. Arch.*, Toronto, Ontario, Canada, Jun. 2017, pp. 1–12.

[43] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," *Queue*, vol. 6, no. 2, pp. 40–53, Mar. 2008.

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 818–833.

[45] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Int. Conf. Mach. Learn.*, Corvalis, OR, USA, Jun. 2007, pp. 759–766.

[46] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 1717–1724.

[47] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Adv. Neural Inform. Process. Syst.*, Montreal, Quebec, Canada, Dec. 2014, pp. 3320–3328.

[48] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017, *arXiv:1610.02357*.

[49] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[51] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, Jul. 2011.

[52] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[53] D. Mishkin and J. Matas, "All you need is a good init," 2015, *arXiv:1511.06422*.

[54] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Int. Conf. Mach. Learn.* Atlanta, GA, USA: JMLR, May 2013, pp. 1139–1147.

[55] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Adv. Neural Inform. Process. Syst.*, vol. 19, no. 1, pp. 153–160, Dec. 2007.

[56] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, Jan. 1990.

[57] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Int. Joint Conf. Neural Netw.*, Washington, DC, USA, Jun. 1989, pp. 593–605.

[58] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[59] H. Specht and E. Lewandowski, "Biased assumptions and oversimplifications in evaluations of citizen science data quality," *Bull. Ecol. Soc. Amer.*, vol. 99, no. 2, pp. 251–256, Apr. 2018.

[60] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[61] W. A. Gardner, "Learning characteristics of Stochastic-Gradient-Descent algorithms: A general study, analysis, and critique," *Sign. Process.*, vol. 6, no. 2, pp. 113–133, Apr. 1984.

[62] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," 2017, *arXiv:1703.00887*.

[63] J. Werfel, X. Xie, and H. S. Seung, "Learning curves for stochastic gradient descent in linear feedforward networks," *Neural Comput.*, vol. 17, no. 12, pp. 2699–2718, Dec. 2005.

[64] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[65] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O (1/k2)," *Sov. Math. Doklady*, vol. 27, no. 2, pp. 372–376, Feb. 1983.

[66] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," 2012, *arXiv:1212.0901*.

[67] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *ACM Int. Conf. Knowl. Disc. Data Mining*, New York, NY, USA, Aug. 2014, pp. 661–670.

[68] N. S. Keskar and R. Socher, "Improving generalization performance by switching from Adam to SGD," 2017, *arXiv:1712.07628*.

[69] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," 2012, *arXiv:1206.1106*.

[70] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[71] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*.

[72] J. Dean *et al.*, "Large scale distributed deep networks," *Adv. Neural Inform. Process. Syst.*, vol. 25, no. 1, pp. 1223–1231, Dec. 2012.

[73] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[74] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," 2017, *arXiv:1708.06020*.

[75] C. Eggert, A. Winschel, and R. Lienhart, "On the benefit of synthetic data for company logo detection," in *ACM Int. Conf. Multimedia*, Brisbane, Australia, Oct. 2015, pp. 1283–1286.

[76] S. Ioffe, “Batch renormalization: Towards reducing minibatch dependence in batch-normalized models,” 2017, *arXiv:1702.03275*.

[77] R. Pascanu, T. Mikolov, and Y. Bengio, “Understanding the exploding gradient problem,” 2012, *arXiv:1211.5063*.

[78] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Int. Conf. Mach. Learn.*, Haifa, IL, USA, Jun. 2010, pp. 807–814.

[79] G. Dahl, T. Sainath, and G. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in *IEEE Int. Conf. Acoust. Speech Sign. Process.*, Vancouver, British Columbia, Canada, May 2013, pp. 8609–8613.

[80] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” 2013, *arXiv:1211.5063*.

[81] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” 2014, *arXiv:1409.5185*.

[82] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Int. Conf. Artif. Intell. and Stat.*, Ft. Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[83] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Eur. Conf. Comput. Vis.* Amsterdam, Netherlands: Springer, Mar. 2016, pp. 630–645.

[84] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 2016, *arXiv:1608.06993*.

[85] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “ShuffleNet V2: Practical guidelines for efficient cnn architecture design,” 2018, *arXiv:1807.11164*.

[86] M. Tan *et al.*, “MnasNet: Platform-aware neural architecture search for mobile,” 2019, *arXiv:1807.11626*.

[87] F. N. Iandola *et al.*, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size,” 2016, *arXiv:1602.07360*.

[88] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks,” in *Adv. Neural Inform. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 4107–4115.

[89] F. Iandola *et al.*, “Densenet: Implementing efficient convnet descriptor pyramids,” 2014, *arXiv:1404.1869*.

[90] S. Zagoruyko and N. Komodakis, “Wide residual networks,” 2016, *arXiv:1605.07146*.

[91] A. G. Howard *et al.*, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, *arXiv:1704.04861*.

[92] S. Beery, D. Morris, and S. Yang, "Efficient pipeline for camera trap image review," 2019, *arXiv:1907.06772*.

[93] L. Liu *et al.*, "Deep learning for generic object detection: A survey," 2019, *arXiv:1809.02165*.

[94] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," 2019, *arXiv:1807.05511*.

[95] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOG-gles: Visualizing object detection features," in *IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 1–8.

[96] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[97] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[98] T. Malisiewicz, A. Gupta, and A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 89–96.

[99] L. Wan, D. Eigen, and R. Fergus, "End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression," 2014, *arXiv:1411.5309*.

[100] W. Ouyang *et al.*, "DeepID-Net: Multi-stage and deformable deep convolutional neural networks for object detection," 2014, *arXiv:1409.3505*.

[101] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" 2015, *arXiv:1502.05082*.

[102] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS – improving object detection with one line of code," in *IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 5561–5569.

[103] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," 2017, *arXiv:1705.02950*.

[104] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 437–446.

[105] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recog.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[106] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Miami, FL, USA, Jun. 2009, pp. 1022–1029.

[107] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *IEEE Conf. Comput. Vis. Pattern Recog.*, New York, NY, USA, Jun. 2006, pp. 37–44.

[108] U. Bonde, V. Badrinarayanan, and R. Cipolla, "Robust instance recognition in presence of occlusion and clutter," in *Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 520–535.

[109] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Adv. Neural Inform. Process. Syst.* Vancouver, British Columbia, Canada: MIT, Dec. 2006, pp. 985–992.

[110] O. Barinova, v. Lempitsky, and P. Kholi, "On detection of multiple object instances using Hough transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1773–1784, Apr. 2012.

[111] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Providence, RI, USA, Jun. 2012, pp. 2578–2585.

[112] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Colorado Springs, CO, USA, Jun. 2011, pp. 617–624.

[113] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, USA, Jun. 2010, pp. 2061–2068.

[114] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, Apr. 2011.

[115] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," 2018, *arXiv:1807.04975*.

[116] G. K. Verma and P. Gupta, "Wild animal detection using deep convolutional neural network," in *Int. Conf. Comput. Vis. Image Process.* Las Vegas, NV, USA: Springer, Aug. 2018, pp. 327–338.

[117] S. Schneider, G. W. Taylor, S. Linquist, and S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data," *Meth. Ecol. Evol.*, vol. 10, no. 4, pp. 461–470, Apr. 2019.

[118] F. Sarwar, A. Griffin, P. Periasamy, K. Portas, and J. Law, "Detecting and counting sheep with a convolutional neural network," in *IEEE Int. Conf. Adv. Video Sign. Based Surveil.*, Auckland, New Zealand, Nov. 2018, pp. 1–6.

[119] T. Trnovszky, P. Kamencay, R. Orjesek, M. Benco, and P. Sykora, "Animal recognition system based on convolutional neural network," *Adv. Elect. Electron. Eng.*, vol. 15, no. 3, pp. 517–525, Sep. 2017.

[120] V. Lopez-Vazquez *et al.*, "Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories," *Sensors*, vol. 20, no. 726, pp. 1–25, Jan. 2020.

[121] N. Rey, M. Volpi, S. Joost, and D. Tuia, "Detecting animals in African Savanna with UAVs and the crowds," *Remote Sens. Environ.*, vol. 200, no. 1, pp. 341–351, Oct. 2017.

[122] C. Vermeulen, P. Lejeune, J. Lisein, P. Sawadogo, and P. Bouché, "Unmanned aerial survey of elephants," *PLoS One*, vol. 8, no. 2, pp. 1–7, Feb. 2013.

[123] J. A. Eikelboom *et al.*, "Improving the precision and accuracy of animal population estimates with aerial image object detection," *Meth. Ecol. Evol.*, vol. 10, no. 11, pp. 1875–1887, Nov. 2019.

[124] I. ŠEvo and A. Avramović, "Convolutional neural network based automatic object detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 740–744, Apr. 2016.

[125] H. Zhu *et al.*, "Orientation robust object detection in aerial images using deep convolutional neural network," in *IEEE Int. Conf. Image Process.*, Quebec City, Quebec, Canada, Sep. 2015, pp. 3735–3739.

[126] F. Sarwar, A. Griffin, S. U. Rehman, and T. Pasang, "Detecting sheep in UAV images," *Comput. Electron. Agriculture*, vol. 187, no. 106219, pp. 2–12, Aug. 2021.

[127] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[128] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.

[129] R. Simpson, K. R. Page, and D. De Roure, "Zooniverse: Observing the world's largest citizen science platform," in *ACM Int. Conf. World Wide Web*, Seoul, Korea, Apr. 2014, pp. 1049–1054.

[130] A. Swanson *et al.*, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna," *Sci. Data*, vol. 2, no. 150026, pp. 1–14, Jun. 2015.

[131] J. P. Cohn, "Citizen science: Can volunteers do real research?" *BioScience*, vol. 58, no. 3, pp. 192–197, Mar. 2008.

[132] A. Irwin, *Citizen Science: A Study of People, Expertise and Sustainable Development*. New York, NY, USA: Routledge, 1995.

[133] J. Silvertown, "A new dawn for citizen science," *Trends Ecol. Evol.*, vol. 24, no. 9, pp. 467–471, Sep. 2009.

[134] G. Van Horn *et al.*, "The iNaturalist species classification and detection dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8769–8778.

[135] M. Haklay, "Geographical citizen science - clash of cultures and new opportunities," in *Workshop Role Volunt. Geo. Inf. in Adv. Sci.*, Zurich, Switzerland, Sep. 2010, pp. 105–122.

[136] N. Kumar *et al.*, "Leafsnap: A computer vision system for automatic plant species identification," in *Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Florence, Italy: Springer, Oct. 2012, pp. 502–516.

[137] B. L. Sullivan *et al.*, "eBird: A citizen-based bird observation network in the biological sciences," *Biol. Conserv.*, vol. 142, no. 10, pp. 2282–2292, Oct. 2009.

[138] M. Ancrenaz, A. Hearn, J. Ross, R. Sollmann, and A. Wilting, *Handbook for Wildlife Monitoring Using Camera-traps*. Kota Kinabalu, Malaysia: BBEC, 2012.

[139] T. Forrester *et al.*, "eMammal – citizen science camera trapping as a solution for broad-scale, long-term monitoring of wildlife populations," in *North Amer. Congr. Conserv. Biol.*, Missoula, MT, USA, Jul. 2014, pp. 80–86.

[140] N. W. Maputla, C. T. Chimimba, and S. M. Ferreira, "Calibrating a camera trap–based biased mark–recapture sampling design to survey the leopard population in the N'wanetsi concession, Kruger National Park, South Africa," *Afr. J. Ecol.*, vol. 51, no. 3, pp. 422–430, Sep. 2013.

[141] M. S. Norouzzadeh *et al.*, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Nat. Acad. of Sci.*, vol. 115, no. 25, pp. 5716–5725, Jun. 2018.

[142] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1431–1439.

[143] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 1884–1892.

[144] S. Wan, Z. Chen, T. Zhang, B. Zhang, and K.-K. Wong, "Bootstrapping face detection with hard negative examples," 2016, *arXiv:1608.02236*.

[145] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 3512–3520.

[146] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Weakly supervised object recognition with convolutional neural networks," in *Adv. Neural Inform. Process. Syst.*, Montreal, Quebec, Canada, Dec. 2014, pp. 1545–5963.

[147] B. Kellenberger, D. Tuia, and D. Morris, "AIDE: Accelerating image-based ecological surveys with interactive machine learning," *Meth. Ecol. Evol.*, vol. 11, no. 12, pp. 1716–1727, Dec. 2020.

[148] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[149] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Feb. 2006.

[150] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *Adv. Neural Inform. Process. Syst.* Denver, CO, USA: MIT, Jan. 1996, pp. 640–646.

[151] Z. Xu, L. Zhu, and Y. Yang, "Few-shot object recognition from machine-labeled web images," 2016, *arXiv:1612.06152*.

[152] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "We don't need no bounding-boxes: Training object class detectors using only human verification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 854–863.

[153] H. G. R. Gouk and A. M. Blake, "Fast sliding window classification with convolutional neural networks," in *ACM Int. Conf. Image Vis. Comput. New Zealand*, Dunedin, New Zealand, Nov. 2014, pp. 114–118.

[154] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Adv. Neural Inform. Process. Syst.*, Lake Tahoe, CA, USA, Dec. 2013, pp. 2553–2561.

[155] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 914–921.

[156] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Mar. 2011.

[157] T. Wang *et al.*, "Detect globally, refine locally: A novel approach to saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3127–3135.

[158] Q. Wang, P. Cavanagh, and M. Green, "Familiarity and pop-out in visual search," *Perception Psychophysics*, vol. 56, no. 5, pp. 495–500, Sep. 1994.

[159] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 2334–2342.

[160] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 391–405.

[161] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 3286–3293.

[162] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[163] S. He, R. W. H. Lau, W. Liu, Z. Huang, and Q. Yang, "SuperCNN: A superpixelwise convolutional neural network for salient object detection," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 330–344, Apr. 2015.

[164] H. Li, H. Lu, Z. Lin, X. Shen, and B. Price, "LCNN: Low-level feature embedded CNN for salient object detection," 2015, *arXiv:1508.03928*.

[165] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Oct. 2015.

[166] H. Jiang *et al.*, "Salient object detection: A discriminative regional feature integration approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Portland, OR, USA, Jun. 2013, pp. 2083–2090.

[167] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Eur. Conf. Comput. Vis.* Amsterdam, Netherlands: Springer, Oct. 2016, pp. 21–37.

[168] X. Li *et al.*, "DeepSaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016.

[169] M. Kümmerer, L. Theis, and M. Bethge, "DeepGaze I: Boosting saliency prediction with feature maps trained on ImageNet," 2014, *arXiv:1411.1045*.

[170] M. Kümmerer, T. S. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," 2016, *arXiv:1610.01563*.

[171] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 678–686.

[172] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, "AttentionNet: Aggregating weak directions for accurate object detection," in *IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 2659–2667.

[173] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 1265–1274.

[174] X. Chu *et al.*, "Multi-context attention for human pose estimation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Honolulu, HI, USA, Jun. 2017, pp. 1831–1840.

[175] B. Wang and D. Klabjan, "An attention-based deep net for learning to rank," 2017, *arXiv:1702.06106*.

[176] M. Spain, "Modeling and predicting object attention in natural scenes," Ph.D. Dissertation, Dept. Comput. Sci., California Inst. of Tech., Pasadena, CA, USA, 2011.

[177] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Salt Lake City, UT, USA, Jun. 2018, pp. 714–722.

[178] K. Hara, M.-Y. Liu, O. Tuzel, and A.-M. Farahmand, "Attentional network for visual object detection," 2017, *arXiv:1702.01478*.

[179] A. R. Kosiorek, A. Bewley, and I. Posner, "Hierarchical attentive recurrent tracking," 2017, *arXiv:1706.09262*.

[180] F. Wang and D. M. J. Tax, "Survey on the attention based RNN model and its applications in computer vision," 2016, *arXiv:1601.06823*.

[181] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368–2378, May 2018.

[182] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, no. 1, pp. 57 120–57 128, Apr. 2019.

[183] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Long Beach, CA, USA, Jun. 2019, pp. 1448–1457.

[184] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Long Beach, CA, USA, Jun. 2019, pp. 3085–3094.

[185] W. Diao *et al.*, "Efficient saliency-based object detection in remote sensing images using deep belief networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 137–141, Jan. 2016.

[186] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Long Beach, CA, USA, Jun. 2019, pp. 8554–8564.

[187] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Long Beach, CA, USA, Jun. 2019, pp. 3917–3926.

[188] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 1201–1210.

[189] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

[190] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014, *arXiv:1311.2524*.

[191] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool, "DeepProposal: Hunting objects by cascading deep convolutional layers," in *IEEE Int. Conf. Comput. Vis.*, Las Condes, Chile, Dec. 2015, pp. 2578–2586.

[192] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," 2017, *arXiv:1612.03144*.

[193] B. Cheng *et al.*, "Revisiting R-CNN: On awakening the classification power of faster R-CNN," 2018, *arXiv:1803.06799*.

[194] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," 2017, *arXiv:1712.00726*.

[195] W. Kuo, B. Hariharan, and J. Malik, "DeepBox: Learning objectness with convolutional networks," in *IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 2479–2487.

[196] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*.

[197] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2016, pp. 516–520.

[198] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[199] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," 2019, *arXiv:1901.01892*.

[200] Z. Shen *et al.*, "DSOD: Learning deeply supervised object detectors from scratch," in *IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 1919–1927.

[201] Y. Li, J. Li, W. Lin, and J. Li, "Tiny-DSOD: Lightweight object detection for resource-restricted usages," 2018, *arXiv:1807.11013*.

[202] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017, *arXiv:1708.02002*.

[203] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-Outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 2874–2883.

[204] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Eur. Conf. Comput. Vis.* Munich, Germany: Springer, Sep. 2018, pp. 385–400.

[205] K. Duan *et al.*, "CenterNet: Keypoint triplets for object detection," 2019, *arXiv:1904.08189*.

[206] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," 2017, *arXiv:1703.07971*.

[207] A. Newell, K. Yang, and J. Deng *et al.*, "Stacked hourglass networks for human pose estimation," in *Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, Netherlands: Springer, Oct. 2016, pp. 483–499.

[208] J. Yang, Q. Liu, and K. Zhang, "Stacked hourglass network for robust facial landmark localisation," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Honolulu, HI, USA, Jun. 2017, pp. 79–87.

[209] L. Yang, J. Liu, and X. Tang, "Object detection and viewpoint estimation with auto-masking neural network," in *Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 441–455.

[210] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," 2015, *arXiv:1506.06204*.

[211] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 297–312.

[212] K. Fragkiadaki, P. Arbeláez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 4083–4090.

[213] H. Hu, S. Lan, Y. Jiang, Z. Cao, and F. Sha, "Fastmask: Segment multi-scale object candidates in one shot," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Honolulu, HI, USA, Jun. 2017, pp. 991–999.

[214] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 321–328.

[215] R. G. Cinbis, J. Verbeek, and C. Schmid, "Segmentation driven object detection with fisher vectors," in *IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 2968–2975.

[216] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.

[217] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[218] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Annu. Conf. Robot Learn.*, Mountain View, CA, USA, Nov. 2017, pp. 1–16.

[219] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robot.* Zurich, Switzerland: Springer, Sep. 2017, pp. 621–635.

[220] W. Qiu and A. Yuille, "UnrealCV: Connecting computer vision to unreal engine," 2016, *arXiv:1609.01326*.

[221] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Adv. Neural Inform. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 379–387.

[222] B. Singh, H. Li, A. Sharma, and L. S. Davis, "R-FCN-3000 at 30fps: Decoupling detection and classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1081–1090.

[223] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Syn. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, Jun. 2009.

[224] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*.

[225] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, USA, Jun. 2010, pp. 2528–2535.

[226] J. Fu *et al.*, "Stacked deconvolutional network for semantic segmentation," 2019, *arXiv:1708.04943*.

[227] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, Netherlands: Springer, Oct. 2016, pp. 354–370.

[228] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Honolulu, HI, USA, Jun. 2017, pp. 472–480.

[229] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*.

[230] *Detectron*. (2018) Accessed: Oct. 29, 2021 [Online]. Available: github.com/facebookresearch/detectron.

[231] J. Brünger, M. Gentz, I. Traulsen, and R. Koch, "Panoptic instance segmentation on pigs," 2020, *arXiv:2005.10499*.

[232] V. Rozsivalova, L. Beran, and I. Hora, "Counting livestock with image segmentation neural network," in *Int. Conf. Soft Comput. Models Ind. Environ. Applicat.* Stockholm, Sweden: Springer Nature, Nov. 2020, pp. 237–244.

[233] A. Singh *et al.*, "Animal detection in man-made environments," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Snowmass Village, CO, USA, Mar. 2020, pp. 1438–1449.

[234] B. Xu *et al.*, "Automated cattle counting using Mask R-CNN in quadcopter vision system," *Comput. Electron. Agriculture*, vol. 171, no. 105300, pp. 1–12, Apr. 2020.

[235] J. G. A. Barbedo, L. V. Koenigkan, T. T. Santos, and P. M. Santos, "A study on the detection of cattle in UAV images using deep learning," *Sensors*, vol. 19, no. 5436, pp. 1–14, Jan. 2019.

[236] X. Zhang *et al.*, "AlignedReID: Surpassing human-level performance in person re-identification," 2017, *arXiv:1711.08184*.

[237] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop Faces 'Real-Life' Imag.: Detect., Align., and Recog.*, E. Learned-Miller, A. Ferencz, and F. Jurie, Eds., Marseille, France, Oct. 2008, pp. 1–14.

[238] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Univ. of Massachusetts, Amherst, Amherst, MA, USA, Tech. Rep. 14.003, 2014.

[239] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Adv. Face Detect. Facial Image Anal.*, vol. 1, M. Kawulok, E. Celebi, and B. Smolka, Eds. Copenhagen, Denmark: Springer Cham, 2016, ch. 8, pp. 189–248.

[240] M. J. Chase *et al.*, "Continent-wide survey reveals massive decline in African Savannah elephants," *PeerJ*, vol. 4, no. 2354, pp. 1–24, Aug. 2016.

[241] P. Juang *et al.*, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *ACM Int. Conf. Arch. Sup. Program. Lang. Oper. Syst.*, San Jose, CA, USA, Oct. 2002, pp. 96–107.

[242] J. G. Mukinya, "An identification method for black rhinoceros (Diceros bicornis Linn. 1758)," *Afr. J. Ecol.*, vol. 14, no. 4, pp. 335–338, Dec. 1976.

[243] K. A. Alexander and M. J. Appel, "African wild dogs (Lycaon pictus) endangered by a canine distemper epizootic among domestic dogs near the Masai Mar National Reserve, Kenya," *J. Wildlife Diseases*, vol. 30, no. 4, pp. 481–485, Oct. 1994.

[244] L. Mech and S. Barber, "A critique of wildlife radio-tracking and its use in national parks: A report to the U.S. National Park Service," U.S. Geological Survey, Northern Prairie Wildlife Research Center, Jamestown, ND, USA, Tech. Rep., 2002.

[245] C. R. Thouless, "Long distance movements of elephants in northern Kenya," *Afr. J. Ecol.*, vol. 33, no. 4, pp. 321–334, Dec. 1995.

[246] M. Lahiri, C. Tantipathananandh, R. Warungu, D. I. Rubenstein, and T. Y. Berger-Wolf, "Biometric animal databases from field photographs: Identification of individual zebra in the wild," in *ACM Int. Conf. Multimedia Retrieval*, Trento, Italy, Apr. 2011, pp. 1–8.

[247] O. O. Patrick, "Demographic status of the Meru elephant population," Kenya Wildlife Service, Nairobi, Kenya, Tech. Rep., 2003.

[248] R. S. Sikes and W. L. Gannon, "Guidelines of the American Society of Mammalogists for the use of wild mammals in research," *J. Mammalogy*, vol. 92, no. 1, pp. 235–253, Feb. 2011.

[249] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification - beyond the capabilities of a human observer," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Snowmass Village, CO, USA, Mar. 2020, pp. 44–52.

[250] A. Polzounov, I. Terpugova, D. Skiparis, and A. Mihai, "Right whale recognition using convolutional neural networks," 2016, *arXiv:1604.05605*.

[251] S. Li, J. Li, H. Tang, R. Qian, and W. Lin, "ATRW: A benchmark for Amur tiger re-identification in the wild," 2019, *arXiv:1906.05586*.

[252] M. Korschens and J. Denzler, "ELPephants: A fine-grained dataset for elephant re-identification," in *IEEE Int. Conf. Comput. Vis. Workshops*, Las Vegas, NV, USA, Oct. 2019, pp. 263–270.

[253] F. Tausch, S. Stock, J. Fricke, and O. Klein, "Bumblebee re-identification dataset," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Snowmass Village, CO, USA, Mar. 2020, pp. 35–37.

[254] S. Beery, G. Van Horn, O. Mac Aodha, and P. Perona, "The iWildCam 2018 challenge dataset," 2019, *arXiv:1904.05986*.

[255] M. H. Khan *et al.*, "AnimalWeb: A large-scale hierarchical dataset of annotated animal faces," 2019, *arXiv:1909.04951*.

[256] J. Nugent, "iNaturalist," *Sci. Scope*, vol. 41, no. 7, pp. 12–13, Mar. 2018.

[257] P. C. Ravoor and T. S. B. Sudarshan, "Deep learning methods for multi-species animal re-identification and tracking - a survey," *Comput. Sci. Rev.*, vol. 38, no. 100289, pp. 1–33, Nov. 2020.

[258] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *Eur. Conf. Comput. Vis.* Florence, Italy: Springer, Oct. 2012, pp. 340–353.

[259] B. G. Weinstein, "A computer vision for animal ecology," *J. Anim. Ecol.*, vol. 87, no. 3, pp. 533–545, May 2018.

[260] G. S. Cheema and S. Anand, "Automatic detection and recognition of individuals in patterned species," in *Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*. Skopje, Macedonia: Springer, Sep. 2017, pp. 27–38.

[261] J. P. Crall, C. v. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, and S. R. Sundaresan, "HotSpotter - patterned species instance recognition," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Tampa, FL, USA, Jan. 2013, pp. 230–237.

[262] H. J. Weideman *et al.*, "Integral curvature representation and matching algorithms for identification of dolphins and whales," in *IEEE Int. Conf. Comput. Vis. Workshops*, Venice, Italy, Oct. 2017, pp. 2831–2839.

[263] O. Moskvyak, F. Maire, A. O. Armstrong, F. Dayoub, and M. Baktashmotlagh, "Robust re-identification of manta rays from natural markings by learning pose invariant embeddings," 2019, *arXiv:1902.10847*.

[264] M. Matthé *et al.*, "Comparison of photo-matching algorithms commonly used for photographic capture–recapture studies," *Ecol. Evol.*, vol. 7, no. 15, pp. 5861–5872, Aug. 2017.

[265] B. Mandal, X. Jiang, H.-L. Eng, and A. Kot, "Prediction of eigenvalues and regularization of eigenfeatures for human face verification," *Pattern Recog. Lett.*, vol. 31, no. 8, pp. 717–724, Jun. 2010.

[266] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with GaussianFace," in *AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 3811–3819.

[267] S. Sengupta *et al.*, "Frontal to profile face verification in the wild," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

[268] N. Ramanathan and R. Chellappa, "Face verification across age progression," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3349–3361, Oct. 2006.

[269] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708.

[270] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, Oct. 2009, pp. 365–372.

[271] G. J. Barord *et al.*, "Comparative population assessments of Nautilus sp. in the Philippines, Australia, Fiji, and American Samoa using baited remote underwater video systems," *PLoS One*, vol. 9, no. 6, pp. 1–5, Jun. 2014.

[272] T. Chehrsimin *et al.*, "Automatic individual identification of Saimaa ringed seals," *IET Comput. Vis.*, vol. 12, no. 2, pp. 146–152, Mar. 2018.

[273] A. Miguel *et al.*, "Identifying individual snow leopards from camera trap images," in *Int. Conf. Sign. Process. Syst.*, vol. 11071. Guangzhou, China: Int. Soc. for Optics and Photonics, May 2019, p. 110710O.

[274] T. A. Morrison, D. Keinath, W. Estes-Zumpf, J. P. Crall, and C. V. Stewart, "Individual identification of the endangered Wyoming toad Anaxyrus baxteri and implications for monitoring species recovery," *J. Herpetology*, vol. 50, no. 1, pp. 44–49, Mar. 2016.

[275] R. B. Nipko, B. E. Holcombe, and M. J. Kelly, "Identifying individual jaguars and ocelots via pattern-recognition software: Comparing HotSpotter and Wild-ID," *Wildlife Soc. Bull.*, vol. 44, no. 2, pp. 424–433, Jun. 2020.

[276] H. Park *et al.*, "Where to spot: Individual identification of leopard cats (Prionailurus bengalensis euptilurus) in South Korea," *J. Ecol. Environ.*, vol. 43, no. 1, pp. 1–5, Dec. 2019.

[277] N. G. Patel and A. Das, "Shot the spots: A reliable field method for individual identification of Amolops formosus (Anura, Ranidae)," *Herpetozoa*, vol. 33, no. 1, pp. 7–16, Oct. 2020.

[278] A. Shukla *et al.*, "A hybrid approach to tiger re-identification," in *IEEE Int. Conf. Comput. Vis. Workshops*, Las Vegas, NV, USA, Oct. 2019, pp. 294–301.

[279] B. G. Weinstein, "MotionMeerkat: Integrating motion video detection and ecological monitoring," *Meth. Ecol. Evol.*, vol. 6, no. 3, pp. 357–362, Mar. 2015.

[280] L. Hiby and P. Lovell, "Computer aided matching of natural markings: A prototype system for grey seals," *Rep. Int. Whaling Commission*, vol. 1, no. Special 12, pp. 3–17, May 1990.

[281] T. Y. Berger-Wolf *et al.*, "Wildbook: Crowdsourcing, computer vision, and data science for conservation," 2017, *arXiv:1710.08880*.

[282] J. M. Lea *et al.*, "Non-invasive physiological markers demonstrate link between habitat quality, adult sex ratio and poor population growth rate in a vulnerable species, the Cape Mountain zebra," *Functional Ecol.*, vol. 32, no. 2, pp. 300–312, Feb. 2018.

[283] A. Oddone, "A mobile application for the image based ecological information system," M.S. Thesis, Dept. Comput. Sci., Univ. of Illinois at Chicago, Chicago, IL, USA, 2016.

[284] S. G. Dunbar *et al.*, "HotSpotter: Less manipulating, more learning, and better vision for turtle photo identification," in *Symp. Sea Turtle Biol. Conserv.*, Las Vegas, NV, USA, Apr. 2017, p. 1.

[285] ——, "HotSpotter: Using a computer-driven photo-id application to identify sea turtles," *J. Exp. Mar. Biol. Ecol.*, vol. 535, no. 151490, pp. 1–11, Feb. 2021.

[286] B. Hughes and T. Burghardt, "Automated visual fin identification of individual great white sharks," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 542–557, May 2017.

[287] D. Blount *et al.*, "Flukebook - continuing growth and technical advancement for cetacean photo identification and data archiving, including automated fin, fluke, and body matching," Int. Whaling Commission, Cambridge, United Kingdom, Tech. Rep. SC/68B/SH/06, 2020.

[288] ——, "Flukebook – recent advances for cetacean photo identification and data archiving including automated fluke matching," Int. Whaling Commission, Cambridge, United Kingdom, Tech. Rep. SC/68A/SH/07, 2019.

[289] J. Calambokidis, J. Barlow, K. Flynn, E. Dobson, and G. H. Steiger, "Update on abundance, trends, and migrations of humpback whales along the US west coast," *IWC Rep. SC A*, vol. 17, no. 1, pp. 1–17, Apr. 2017.

[290] T. Franklin *et al.*, "Photo-identification of individual southern hemisphere humpback whales (Megaptera novaeangliae) using all available natural marks: Managing the potential for misidentification," *J. Cetacean Res. Manage.*, vol. 21, no. 1, pp. 71–83, Oct. 2020.

[291] H. J. Weideman, "Contour-based instance recognition of animals," Ph.D. Dissertation, Dept. Comput. Sci., Rensselaer Polytechnic Inst., Troy, NY, USA, 2019.

[292] P. Kulits, J. Wall, A. Bedetti, M. Henley, and S. Beery, "ElephantBook: A semi-automated human-in-the-loop system for elephant re-identification," 2021, *arXiv:2106.15083*.

[293] R. Bogucki *et al.*, "Applying deep learning to right whale photo identification," *Conserv. Biol.*, vol. 33, no. 3, pp. 676–684, Jun. 2019.

[294] A. Kabani and M. R. El-Sakka, "Improving right whale recognition by fine-tuning alignment and using wide localization network," in *IEEE Can. Conf. Elect. Comput. Eng.*, Windsor, Ontario, Canada, Aug. 2017, pp. 1–6.

[295] ——, "North Atlantic right whale localization and recognition using very deep and leaky neural network," *Math. Appl.*, vol. 5, no. 2, pp. 155–170, Nov. 2016.

[296] B. M. Norman, S. Reynolds, and D. L. Morgan, "Does the whale shark aggregate along the western Australian coastline beyond Ningaloo Reef?" *Pacific Conserv. Biol.*, vol. 22, no. 1, pp. 72–80, Apr. 2016.

[297] G. Araujo *et al.*, "Getting the most out of citizen science for endangered species such as whale shark," *J. Fish Biol.*, vol. 96, no. 4, pp. 864–867, Apr. 2020.

[298] J. A. McKinney *et al.*, "Long-term assessment of whale shark population demography and connectivity using photo-identification in the western Atlantic ocean," *PLoS One*, vol. 12, no. 8, pp. 1–18, Aug. 2017.

[299] A. Batbouta, "Computer assisted labeling of humpback whales and whale sharks," M.S. Thesis, Dept. Comput. Sci., Rensselaer Polytechnic Inst., Troy, NY, USA, 2019.

[300] W. Winckler *et al.*, "Comparison of fine-scale recombination rates in humans and chimpanzees," *Sci.*, vol. 308, no. 5718, pp. 107–111, Apr. 2005.

[301] T. Burghardt, J. Calic, and B. T. Thomas, "Tracking animals in wildlife videos using face detection," in *EWIMT*, London, United Kingdom, Nov. 2004, pp. 1–8.

[302] T. Burghardt and J. ĆAlić, "Analysing animal behaviour in wildlife videos using face detection and tracking," *IEEE Vis. Image Sign. Process.*, vol. 153, no. 3, pp. 305–312, Jun. 2006.

[303] D. Deb *et al.*, "Face recognition: Primates in the wild," in *IEEE Int. Conf. Biomet. Theor. Applicat. Syst.*, Miyazaki, Japan, Oct. 2018, pp. 1–10.

[304] A. Freytag *et al.*, "Chimpanzee faces in the wild: Log-Euclidean CNNs for predicting identities and attributes of primates," in *German Conf. Pattern Recog.* Hannover, Germany: Springer, Sep. 2016, pp. 51–63.

[305] D. Schofield *et al.*, "Chimpanzee face recognition from videos in the wild using deep learning," *Sci. Adv.*, vol. 5, no. 9, pp. 1–9, Sep. 2019.

[306] D. Crouse *et al.*, "LemurFaceID: A face recognition system to facilitate individual identification of lemurs," *BMC Zoology*, vol. 2, no. 1, pp. 1–14, Dec. 2017.

[307] M. Clapham, E. Miller, M. Nguyen, and C. T. Darimont, "Automated facial recognition for wildlife that lack unique markings: A deep learning approach for brown bears," *Ecol. Evol.*, vol. 10, no. 23, pp. 12 883–12 892, Dec. 2020.

[308] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *IEEE Int. Conf. Image Process.*, Rochester, NY, USA, Sep. 2002, pp. 900–903.

[309] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *IEEE Int. Conf. Comput. Vis.*, vol. 2, Beijing, China, Oct. 2005, pp. 1619–1626.

[310] M. Kerr, "Facebook for the ferocious," *Sci. Amer.*, vol. 313, no. 1, pp. 21–21, Jun. 2015.

[311] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Eur. Conf. Comput. Vis.* Munich, Germany: Springer, Sep. 2018, pp. 459–474.

[312] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[313] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 2015, pp. 815–823.

[314] H. Weideman *et al.*, "Extracting identifying contours for African elephants and humpback whales using a learned appearance model," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Snowmass Village, CO, USA, Mar. 2020, pp. 1276–1285.

[315] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Miami, FL, USA, Jun. 2009, pp. 9–16.

[316] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lisbon, Portugal, Feb. 2009, pp. 331–340.

[317] S. McCann and D. G. Lowe, "Local naive Bayes nearest neighbor for image classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Providence, RI, USA, Jun. 2012, pp. 3650–3656.

[318] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[319] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[320] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Jan. 2005.

[321] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Long Beach, CA, USA, Jun. 2019, pp. 1920–1929.

[322] J. W. Thompson *et al.*, "finFindR: Computer-assisted recognition and identification of bottlenose dolphin photos in R," 2019, *bioRxiv:825661*.

[323] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[324] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," 2014, *arXiv:1406.2952*.

[325] E. Nepovinnykh, T. Eerola, and H. Kalviainen, "Siamese network based pelage pattern matching for ringed seal re-identification," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Snowmass Village, CO, USA, Mar. 2020, pp. 25–34.

[326] N. Dlamini and T. L. Van Zyl, "Automated identification of individuals in wildlife population using siamese neural networks," in *IEEE Int. Conf. Soft Comput. Mach. Intell.*, Stockholm, Sweden, Nov. 2020, pp. 224–228.

[327] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, Jun. 2016, pp. 378–383.

[328] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Eur. Conf. Comput. Vis.* Amsterdam, Netherlands: Springer, Oct. 2016, pp. 791–808.

[329] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, "A siamese long short-term memory architecture for human re-identification," in *Eur. Conf. Comput. Vis.* Amsterdam, Netherlands: Springer, Oct. 2016, pp. 135–153.

[330] C. G. J. Petersen, "The yearly immigration of young plaice into the Limfjord from the German Sea," *Rep. Danish Biol. Station*, vol. 6, no. 1, pp. 1–48, Jan. 1896.

[331] T. Y. Berger-Wolf *et al.*, "IBEIS: Image-based ecological information system: From pixels to science and conservation," in *Bloomberg Data for Good Exchange Conf.*, New York, NY, USA, Sep. 2015, pp. 1–4.

[332] D. Chapman and N. Chapman, *Fallow Deer: Their History, Distribution, and Biology*. Ithaca, NY, USA: Dalton, 1975.

[333] K. U. Karanth, N. S. Kumar, V. R. Goswami, J. D. Nichols, and S. Hedges, "Estimating abundance and other demographic parameters in elephant populations using capture–recapture sampling: Field practices," in *Monitoring Elephant Populations and Assessing Threats: A Manual for Researchers, Managers and Conservationists*, vol. 1, S. Hedges, Ed. Himayatnagar, Hyderabad, India: Univ. Press, 2012, ch. 10, pp. 172–213.

[334] G. C. White and K. P. Burnham, "Program MARK: Survival estimation from populations of marked animals," *Bird Stud.*, vol. 46, no. Supplement 001, pp. 120–139, Jan. 1999.

[335] S. W. Pacala and J. Roughgarden, "Population experiments with the Anolis lizards of St. Maarten and St. Eustatius," *Ecol.*, vol. 66, no. 1, pp. 129–141, Feb. 1985.

[336] S. T. Buckland and P. H. Garthwaite, "Quantifying precision of mark-recapture estimates using the bootstrap and related methods," *Biometrics*, vol. 47, no. 1, pp. 255–268, Mar. 1991.

[337] D. T. Bolger, T. A. Morrison, B. Vance, D. Lee, and H. Farid, "A computer-assisted system for photographic mark–recapture analysis," *Meth. Ecol. Evol.*, vol. 3, no. 5, pp. 813–822, Oct. 2012.

[338] L. Hiby *et al.*, "Analysis of photo-id data allowing for missed matches and individuals identified from opposite sides," *Meth. Ecol. Evol.*, vol. 4, no. 3, pp. 252–259, Mar. 2013.

[339] G. M. Jolly, "Explicit estimates from capture-recapture data with both death and immigration-stochastic model," *Biometrika*, vol. 52, no. 1/2, pp. 225–247, Jun. 1965.

[340] G. A. Seber, "A note on the multiple-recapture census," *Biometrika*, vol. 52, no. 1/2, pp. 249–259, Jun. 1965.

[341] F. M. Zanzotto, "Viewpoint: Human-in-the-loop artificial intelligence," *J. Artif. Intell. Res.*, vol. 64, no. 1, pp. 243–252, Feb. 2019.

[342] D. Xin *et al.*, "Accelerating human-in-the-loop machine learning: Challenges and opportunities," in *ACM 2nd Workshop Data Manage. End-to-end Mach. Learn.*, New York, NY, USA, Jun. 2018, pp. 1–4.

[343] ——, "Helix: Accelerating human-in-the-loop machine learning," *VLDB Endowment*, vol. 11, no. 12, pp. 1958–1961, Aug. 2018.

[344] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan, "Human decisions and machine predictions," *Quart. J. Econ.*, vol. 133, no. 1, pp. 237–293, Feb. 2018.

[345] H. Jiang and O. Nachum, "Identifying and correcting label bias in machine learning," in *Int. Conf. Artif. Intell. and Stat.*, vol. 108. Palermo, Italy: PMLR, Aug. 2020, pp. 702–712.

[346] J. Gu and D. Oelke, "Understanding bias in machine learning," 2019, *arXiv:1909.01866*.

[347] S. Schelter and J. Stoyanovich, "Taming technical bias in machine learning pipelines," *Database Eng. Bull.*, vol. 43, no. 4, pp. 39–50, Dec. 2020.

[348] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, May 1994.

[349] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 129–145, Mar. 1996.

[350] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. of Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.

[351] S. Oh, A. Ashiquzzaman, D. Lee, Y. Kim, and J. Kim, "Study on human activity recognition using semi-supervised active transfer learning," *Sensors*, vol. 21, no. 2760, pp. 1–14, Jan. 2021.

[352] W. Lindskog, "Time series active learning using automated feature extraction," M.S. Thesis, Dept. Math. Sci., Lund Univ., Lund, Sweden, 2021.

[353] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: A survey," *VLDB J.*, vol. 29, no. 1, pp. 93–117, Jan. 2020.

[354] E. Rezig *et al.*, "Dagger: A data (not code) debugger," in *Conf. Innov. Data Syst. Res.*, Amsterdam, Netherlands, Jan. 2020, pp. 12–15.

[355] E. K. Rezig *et al.*, "Data Civilizer 2.0: A holistic framework for data preparation and analytics," *VLDB Endowment*, vol. 12, no. 12, pp. 1954–1957, Aug. 2019.

[356] T. Berger-Wolf *et al.*, "The Great Grevy's Rally: The need, methods, findings, implications and next steps," Grevy's Zebra Trust, Nairobi, Kenya, Tech. Rep., 2016.

[357] D. Rubenstein *et al.*, "The state of Kenya's Grevy's zebras and reticulated giraffes: Results of the Great Grevy's Rally 2018," Grevy's Zebra Trust, Nairobi, Kenya, Tech. Rep., 2018.

[358] D. I. Rubenstein, "Ecology, social behavior, and conservation in zebras," in *Behav. Ecol. Trop. Anim.*, (Adv. Stud. Behav.), vol. 42, R. Macedo, Ed. Burlington, MA, USA: Elsevier, 2010, ch. 7, pp. 231–258.

[359] *Lasagne*. (2015) Accessed: Oct. 29, 2021 [Online]. Available: github.com/Lasagne/Lasagne.

[360] J. Bergstra *et al.*, "Theano: a CPU and GPU math expression compiler," in *Python Sci. Comput. Conf.*, vol. 4, Austin, TX, USA, Jun. 2010, pp. 3–10.

[361] F. Bastien *et al.*, "Theano: New features and speed improvements," 2012, *arXiv:1211.5590*.

[362] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, Vancouver, British Columbia, Canada, Dec. 2019, pp. 8024–8035.

[363] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 675–678.

[364] *Lightnet*. (2018) Accessed: Oct. 29, 2021 [Online]. Available: gitlab.com/EAVISE/lightnet.

[365] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 259–289, May 2008.

[366] *OpenMP*. (2008) Accessed: Oct. 29, 2021 [Online]. Available: openmp.org/mp-documents/spec30.pdf.

[367] K. E. Van De Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 1879–1886.

[368] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," in *Ann. IEEE/ACM Int. Symp. Microarch.*, Washington, DC, USA, Dec. 2012, pp. 449–460.

[369] B. Graham, "Spatially-sparse convolutional neural networks," 2014, *arXiv:1409.6070*.

[370] A. Zlateski, K. Lee, and H. S. Seung, "ZNN – a fast and scalable algorithm for training 3D convolutional networks on multi-core and many-core shared memory machines," in *IEEE Int. Parallel Distrib. Process. Symp.*, Chicago, IL, USA, May 2016, pp. 801–811.

[371] G. Caughley, "Sampling in aerial survey," *J. Wildlife Manage.*, vol. 41, no. 4, pp. 605–615, Oct. 1977.

[372] H. Jachmann, "Comparison of aerial counts with ground counts for large African herbivores," *J. Appl. Ecol.*, vol. 39, no. 5, pp. 841–852, Oct. 2002.

[373] G. Melville, J. Tracey, P. Fleming, and B. Lukins, "Aerial surveys of multiple species: Critical assumptions and sources of bias in distance and mark–recapture estimators," *Wildlife Res.*, vol. 35, no. 4, pp. 310–348, Jun. 2008.

[374] A. Graham and R. Bell, "Investigating observer bias in aerial survey by simultaneous double-counts," *J. Wildlife Manage.*, vol. 53, no. 4, pp. 1009–1016, Oct. 1989.

[375] C. D. Becker and J. R. Ginsberg, "Mother-infant behaviour of wild Grevy's zebra: Adaptations for survival in semidesert East Africa," *Anim. Behav.*, vol. 40, no. 6, pp. 1111–1118, Dec. 1990.

[376] C. G. J. Petersen and P. Boysen-Jensen, *Valuation of the Sea*. Copenhagen, Denmark: Centraltrykkeriet, 1911.

[377] G. M. Jolly and J. M. Dickson, "The problem of unequal catchability in mark–recapture estimation of small mammal populations," *Can. J. Zoology*, vol. 61, no. 4, pp. 922–927, Apr. 1983.

[378] J. O. Ogutu *et al.*, "Changing wildlife populations in Nairobi National Park and adjoining Athi-Kaputiei plains: Collapse of the migratory wildebeest," *Open Conserv. Biol. J.*, vol. 7, no. 1, pp. 11–26, Jul. 2013.

[379] S. Ngene *et al.*, "Total aerial count of elephants, Grevy'szebra and other large mammals in Laikipia-Samburu-Marsabit ecosystem in (November 2012)," Kenya Wildlife Service, Nairobi, Kenya, Tech. Rep., 2013.

[380] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

[381] D. I. Rubenstein *et al.*, "The Great Zebra and Giraffe Count: The power and rewards of citizen science," Kenya Wildlife Service, Nairobi, Kenya, Tech. Rep., 2015.

APPENDIX A
GGR 2018 PARTICIPANT GUIDE

A.1 Welcome to the 2018 Great Grévy's Rally

On the 27th and 28th of January 2018, you will be part of a historic event that will take place in northern Kenya. Within 45 censusing blocks spread across the country, scientists, local community members, and citizens will drive, fly, and take photographs. The area that you will help to cover consists of the core range for the Grévy's zebra and reticulated giraffe, which are the two target species of the Rally. Only in Kenya has such an extensive census by the public, covering over 25,000 KM2, ever been completed.

Our team pioneered such a massive effort during the first Great Grévy's Rally in 2016, when it was estimated that 2,350 Grévy's zebra roamed the semi-arid regions of Kenya. This year we are expanding the Rally to 1) include more participants and 2) add the reticulated giraffe, a species the IUCN has recently listed as threatened. The tens - or hundreds - of thousands of pictures that you will help to collect over the next two days will be uploaded to Wildbook and given to image analysis algorithms. Our algorithms work to identify each unique animal by performing a "sight-resight" analysis, a non-invasive variant of the traditional "mark-recapture" techniques used by population biologists. Using the visual appearance of Grévy's zebras with their naturally barcoded stripes and the distinctive polygon patterns on reticulated giraffes, scientists will be able to estimate the size of animal populations within each county and across the entire nation. We will also use your pictures to determine the age and sex of the animals so that we can estimate the health and sustainability of the population. By being a volunteer for the Rally, your contributions will go directly towards protecting these animals and ensuring the future of their species.

For the Great Grévy's Rally to succeed in 2018, it is essential that each participant follow some simple rules and guidelines. This document will introduce you to your GGR camera (and camera supplies) and provide examples of pictures to take.

A.2 Hardware Tote Bag

Your tote bag contains the following:

- A **plastic camera bag with**:
  - A Nikon COOLPIX S9900 digital camera, with a carrying case

  - USB charging cable and base charger
  - A Kenyan plug adaptor OR USB car charger
  - A CAMERA ID CARD (with the GGR logo, QR code, a number and letter).

The CAMERA ID CARDS are used to organize all of the different cameras and their images during the Rally. You will be asked to take a photograph of your CAMERA ID CARD at the beginning of each day, synchronized with all other cameras in your vehicle (if any), so please make sure it is not misplaced.

- This **participant guide packet**, which includes:
  - **Introduction to your GGR Camera**
  - **Use of Personal Digital Cameras**
  - **Rally Day Start** - *the procedure to follow at the start of each day*
  - **Turning on Your GGR Camera's GPS Function**
  - **Examples of Good and Bad Pictures** - *a green/red reference sheet*

A.3 Introduction to Your Camera

Each vehicle has been assigned a single, numbered GGR camera with built-in GPS.

DO NOT remove the camera battery. The time and date settings have been set on the camera to Kenyan time. If the battery is removed, then the camera will revert to default settings. If the time and date need to be reset, refer to page 13 of the camera manual.

The D-Pad is the circle directly above the MENU button with four icons and an OK button in the center.

To prepare for your first day of the Rally, take the following steps:

1. Make sure the satellite icon is visible on the bottom left of the display (above the battery icon), which indicates that the GPS function has been turned on. A red satellite icon means that the camera has not found sufficient GPS satellites to function correctly. Take the camera outside with a clear view of the sky and wait 5 minutes to acquire the GPS signals.

2. When the camera is fully connected, white boxes will appear next to the satellite icon. See the guidelines below:

Location data reception can be checked on the shooting screen.

- or : Signals are being received from four or more satellites, or from three satellites, and positioning is being performed.
- : Signals are being received, but positioning is not possible.
- : Signals are not received.

Figure A. 1: GGR-18 participant guide, image 1.

If there is NO satellite icon on the display, follow the below instructions for Turning on Your GGR Camera's GPS Function

A.4 Use of Personal Digital Cameras During the Rally

If you are a passenger in a GGR vehicle and want to use a personal digital camera, please follow the instructions below. Once these steps are complete, you can continue with the **Rally Day Start** procedure just like other GGR cameras.

1. Ensure your camera is fully charged and bring extra batteries, if possible.
2. Set the time and date to Kenyan time (GMT+3)
3. Shoot in JPEG mode only (no RAW). Pictures from film cameras can not be used.
4. If your camera has a Digital Zoom function, turn it off or only use optical zoom
5. Obtain a CAMERA ID CARD with the letter B, C, D, E, or F from your GGR driver. Be sure to take a photo of your card at exactly the same time as your driver takes a picture of their card. These simultaneous pictures will sync your pictures with the GPS records that the driver's GGR camera will create.

A.5 Rally Day Start

Start Here Each Morning of the Rally

A.5.1 Start of the Day's Rally - Start GPS Log

1. *If you are using a personal camera, continue to Step (2) on next page.*
2. Press the MENU button to open the camera menu
3. Push left on the D-Pad to select a menu page
4. Push down on the D-Pad 3 times to highlight the GPS satellite icon
5. Push the OK button
6. Push down on the D-Pad 3 times to highlight **Create Log**
7. Push the OK button. Your screen should look like this:

Figure A. 2: GGR-18 participant guide, image 2.

8. If the **Log Interval** is already set to 10s, skip to step 12.
9. Push down on the D-Pad 1 time to highlight **Log Interval**
10. Push the OK button. Your screen should look like this:

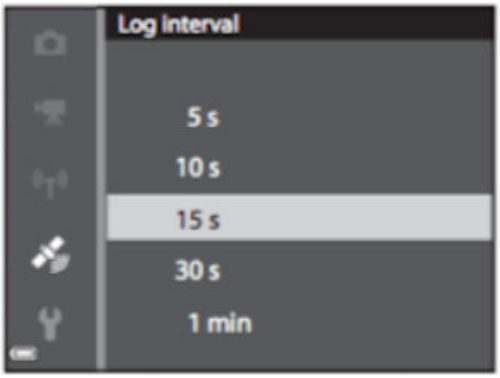


Figure A. 3: GGR-18 participant guide, image 3.

11. Push up on the D-Pad 1 time to highlight **10s**
12. Push the OK button
13. Push up on the D-Pad 1 time to highlight **Start Log**
14. Push the OK button
15. Push down on the D-Pad 1 time to highlight **Log data for next 12 hrs**
16. Push the OK button
17. Your screen should look like this:

Figure A. 4: GGR-18 participant guide, image 4.

18. *Now your tracking log has started!*

19. Verify that the highlighted menu item at the bottom says **End log** and that the duration and interval numbers are correct

    (a) **NOTE: DO NOT** select **End log** (this is done at the end of the day)

20. Press the MENU button to close the camera menu

21. The satellite icon on the bottom left of display (above the battery icon) should display the word **LOG** to indicate that GPS logging is turned on

A.5.2 Take a Synchronized Picture of All Camera ID Cards

1. At the same time, all cameras in the vehicle will take a single, synchronized picture of their assigned CAMERA ID CARD. Each camera should take a picture of their assigned card at the start of every day.

2. Begin a countdown for this picture by having the driver say, out loud, "3-2-1 SNAP!". A reminder to take this picture is shown at the bottom of the card.

3. The GGR camera should always have a CAMERA ID CARD with the letter A

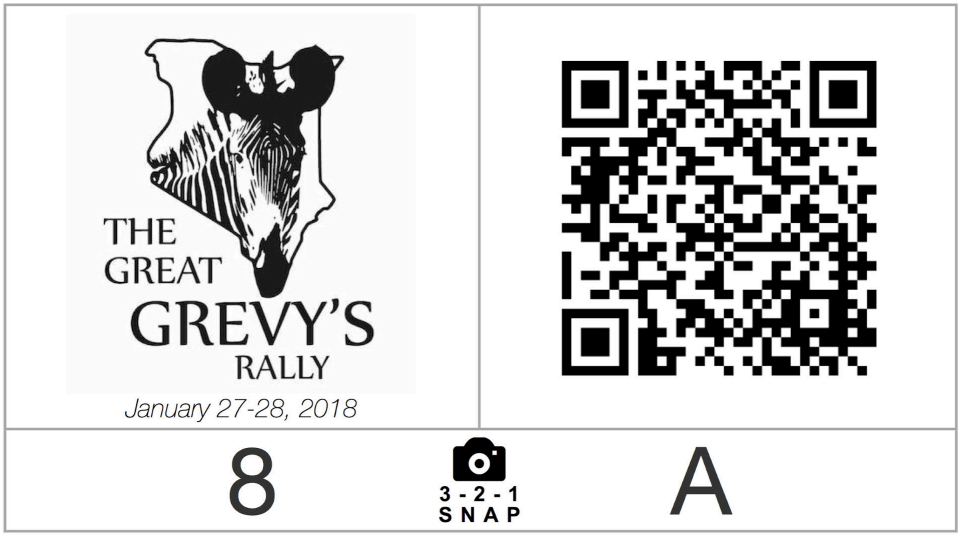


Figure A. 5: GGR-18 participant guide, image 5.

4. If there is more than one photographer in a vehicle, each camera is assigned a CAMERA ID CARD. All additional cameras should use the letters B, C, D, E, or F, with A being reserved

for the (only or primary) GGR camera in the car.

5. Save your CAMERA ID CARD for it is needed at the start of each day of the Rally.

A.5.3 Start your Rally!

Go drive (or fly) to find Grévy's zebras and reticulated giraffes in the wild!

A.5.4 Taking Pictures of Grévy's Zebras and Reticulated Giraffes

1. Photographing the animals

   (a) **Group Photos** - As you approach, take a photo of the Grévy's zebra herd or the tower of giraffes from a distance. Try to capture the entire herd using only 1 or 2 images. Do this even if a single Grévy's zebra or reticulated giraffe is seen alone.

   (b) **Individual Photos** - After you have taken the group photo from a distance, approach the group of animals to get a closer look and take photos of each individual animal.

2. **Target one animal at a time. Take a picture of only the RIGHT side of the animal (facing or walking to the right).** To ensure the best possible chance of identifying the animal, try to put the animal in the **CENTER** of the image and try to photograph the entire animal (not just a piece of it). Refer to the green/red reference sheet for good and bad examples.

3. Repeat Step 2 for each Grévy's zebra or reticulated giraffe seen in the group. You should aim to take 3-4 different RIGHT side pictures of each zebra or giraffe as it moves around. Don't worry if an animal moves away or behind something (bushes or other animals) while taking photographs – the most important thing is to get at least one RIGHT side photograph for each animal in the group.

4. *Be patient!* The animals will be curious about you and will move around slowly. Try to wait for when you can clearly see the RIGHT side of the animal without harsh sunlight glare. Avoid taking pictures of animals that are significantly behind bushes or other animals. You may need to relocate yourself to find a better spot to get good, RIGHT side photos of individuals.

5. If after a while you cannot get any pictures of the RIGHT side or if the animals start to run off, then take any picture possible.

Remember: "Right is Right!"

Only take pictures of the RIGHT side of a zebra or giraffe.

A.5.5 End of the Day's Rally - End GPS Log

1. **If you are using a personal camera, continue to Step (6) at the bottom of page**
2. Press the MENU button to open the camera menu
3. Push left on the D-Pad to select a menu page
4. Push down on the D-Pad 3 times to highlight the GPS satellite icon
5. Push the OK button
6. Push the OK button to highlight menu item **Create log**
7. Push the OK button
8. Your screen should look like the image below. Push the OK button on the (already) highlighted menu item End log.

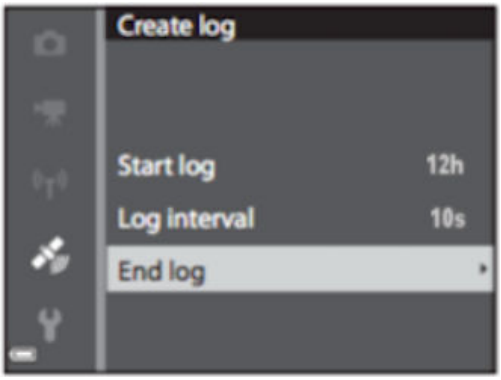


Figure A. 6: GGR-18 participant guide, image 6.

9. Your screen should look like the image below. Push the OK button on the (already) highlighted menu item **Save log**

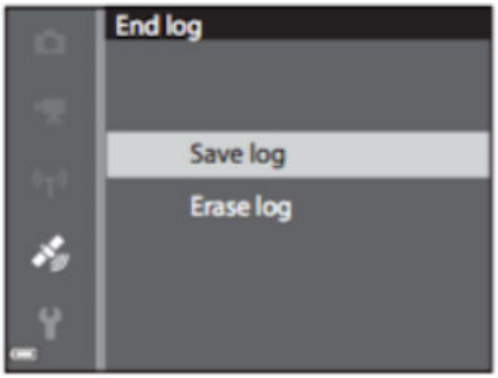


Figure A. 7: GGR-18 participant guide, image 7.

10. This operation could take a few minutes to complete, **DO NOT** turn off the camera during this process. The screen may or may not change while it is saving the location log to the SD card. Once it is complete, a message will appear across the screen saying the log has been saved.
11. Push the OK button
12. Press the MENU button again to close the camera menu

A.5.6 Prepare for Day 2 of the Rally

1. Turn off your camera
2. Charge your camera batteries overnight
3. Save your CAMERA ID CARD. You will need to take a synchronization picture of your card at the start of the second day of the Rally.

A.6 Turning on Your GGR Camera's GPS Function

1. Press the MENU button on the bottom right of the back of the camera (next to the display) to open the camera menu
2. Push left on the D-Pad to select a menu page
3. Push down on the D-Pad 3 times to highlight the GPS satellite icon
4. Push the OK button (located in the center of the D-Pad)

5. Push the OK button on the (already) highlighted men item **Location data options**
6. Push the OK button on the (already) highlighted menu item **Record location data...OFF**
7. Push up on the D-Pad to highlight On next to the satellite icon, as seen in the screen below:

Figure A. 8: GGR-18 participant guide, image 8.

8. Push the OK button
9. Verify that the menu item says Record location data. . . ON
10. Press the MENU button again to close the camera menu
11. A satellite icon should appear on the bottom left of the display (above the battery icon), which indicates that the GPS function has been turned on. A red satellite icon means that the camera has not found sufficient GPS satellites to function correctly. Take the camera outside with a clear view of the sky and wait 5 minutes to acquire the GPS signals.
12. Once the camera is fully connected, white boxes will load next to the satellite icon. See the guidelines below:

Location data reception can be checked on the shooting screen.

- or : Signals are being received from four or more satellites, or from three satellites, and positioning is being performed.
- : Signals are being received, but positioning is not possible.
- : Signals are not received.

Figure A. 9: GGR-18 participant guide, image 9.

APPENDIX B

CHAPTER ATTRIBUTIONS & COPYRIGHT PERMISSIONS

B.1 Copyright Permissions for Chapter 1

J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

File Name	"Parham 2016 WACV - Detecting Plains and Grevy's Zebras in the Real World - Copyright.pdf"
Copyright	IEEE 2016
File Type	Portable Document Format (PDF)
File Size	66 KB

J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

File Name	"Parham 2017 SSS AISOC Paper - Animal Population Censusing at Scale with Citizen Science and Photographic Identification - Copyright.pdf"
Copyright	AAAI 2017
File Type	Portable Document Format (PDF)
File Size	120 KB

B.2 Copyright Permissions for Chapter 2

J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

File Name	"Parham 2016 WACV - Detecting Plains and Grevy's Zebras in the Real World - Copyright.pdf"
Copyright	IEEE 2016
File Type	Portable Document Format (PDF)
File Size	66 KB

J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

File Name	"Parham 2017 SSS AISOC Paper - Animal Population Censusing at Scale with Citizen Science and Photographic Identification - Copyright.pdf"
Copyright	AAAI 2017
File Type	Portable Document Format (PDF)
File Size	120 KB

J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

File Name	"Parham 2018 WACV - An Animal Detection Pipeline for Identification - Copyright.pdf"
Copyright	IEEE 2018
File Type	Portable Document Format (PDF)
File Size	67 KB

B.3 Copyright Permissions for Chapter 3

J. Parham and C. Stewart, "Detecting plains and Grevy's zebras in the real world," in *IEEE Winter Conf. Applicat. Comput. Vis. Workshops*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

File Name	"Parham 2016 WACV - Detecting Plains and Grevy's Zebras in the Real World - Copyright.pdf"
Copyright	IEEE 2016
File Type	Portable Document Format (PDF)
File Size	66 KB
Tables	3.2, 3.3
Figures	3.1, 3.3, 3.9, 3.10, 3.11

J. Parham *et al.*, "An animal detection pipeline for identification," in *IEEE Winter Conf. Applicat. Comput. Vis.*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1–9.

File Name	"Parham 2018 WACV - An Animal Detection Pipeline for Identification - Copyright.pdf"
Copyright	IEEE 2018
File Type	Portable Document Format (PDF)
File Size	67 KB
Tables	3.1
Figures	3.2, 3.4, 3.8, 3.12, 3.13, 3.14, 3.16, 3.17, 3.21, 3.22

B.4 Copyright Permissions for Chapter 6

J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. I. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *AAAI Spring Symp.*, Palo Alto, CA, USA, Jan. 2017, pp. 37–44.

File Name	"Parham 2017 SSS AISOC Paper - Animal Population Censusing at Scale with Citizen Science and Photographic Identification - Copyright.pdf"
Copyright	AAAI 2017
File Type	Portable Document Format (PDF)
File Size	120 KB
Tables	6.1, 6.2
Figures	6.5, 6.9, 6.6, 6.10, 6.17, 6.18

J. Parham, C. Stewart, T. Berger-Wolf, D. Rubenstein, and J. Holmberg, "The Great Grevy's Rally: A review on procedure," in *AI Wildlife Conserv. Workshop*, Stockholm, Sweden, Jul. 2018, pp.1–3.

File Name	"Parham 2018 ICJAI-AIWC 2018 - The Great Grevy's Rally - A Review on Procedure - Copyright.pdf"
Copyright	IJCAI 2018
File Type	Portable Document Format (PDF)
File Size	196 KB
Tables	6.1
Figures	6.7, 6.8