Interactive Machine Vision for Wildlife Conservation

Benjamin Kellenberger

Interactive Machine Vision for Wildlife Conservation

Benjamin Kellenberger

Propositions

1. Wildlife conservation strongly profits from machine vision-assisted population censuses.
   (this thesis)

2. Machine learning algorithms yield better results if integrated with humans in a feedback loop.
   (this thesis)

3. The comfort of familiar paradigms is one of the main reasons why many research fields don't advance as fast as they could.

4. The academization of non-scientific professions only leads to inappropriate qualification requirements.

5. Every student should attend a software engineering course as early as possible in their career.

6. Enforced societal equality is the biggest threat to equity.

Propositions belonging to the thesis entitled:

"Interactive Machine Vision for Wildlife Conservation"

Benjamin Kellenberger
Wageningen, April 6, 2020

Interactive Machine Vision for Wildlife Conservation

Benjamin Kellenberger

Thesis committee

Promotor:
Prof. Dr D. Tuia
Professor of Geo-information Sciences
Wageningen University & Research
Other members:
Prof. Dr F. van Langevelde, Wageningen University & Research
Dr S. Zuffi, Consiglio Nazionale della Ricerche, Milan, Italy
Dr T. Burghardt, University of Bristol, United Kingdom
Prof. Dr T. Berger-Wolf, The Ohio State University, United States of America

This research was conducted under the auspices of the C.T. de Wit Graduate School of Production Ecology & Resource Conservation (PE&RC)

Interactive Machine Vision for Wildlife Conservation

Benjamin Kellenberger

Thesis
submitted in fulfilment of the requirements for the degree of doctor at
Wageningen University
by the authority of the Rector Magnificus
Prof. Dr A.P.J. Mol,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on April 6 2020
at 4 p.m. in the Aula.

Benjamin Kellenberger
Interactive Machine Vision for Wildlife Conservation

154 pages

PhD thesis, Wageningen University, Wageningen, NL (2020)
With references, with summary in English and German

ISBN 978-94-6395-273-6
DOI https://doi.org/10.18174/511122

Summary

The loss rate of endangered animal species has reached levels that are critical enough for our time to be called the sixth mass extinction. Families of vertebrates and large mammals, such as Rhinocerotidae, are likely to become extinct in a few years unless countermeasures are taken. Before doing so, however, it is imperative to assess current animal population sizes through wildlife censuses. Furthermore, conservation efforts require animal populations to be monitored over time, which implies conducting census repetitions over multiple years.

Recent developments in technology have paved the way for animal census efforts of unprecedented accuracies and scales, predominantly through the employment of Unmanned Aerial Vehicles (UAVs). UAVs allow for acquiring aerial imagery of vast areas over *e.g.* a wildlife reserve, and thereby provide evidence of the abundance and location of individuals in a safe manner. Hitherto, the main challenge of UAV-enforced animal censuses has been the stage of manual photo-interpretation, in which animals have to be tediously identified and annotated by hand in potentially tens of thousands of aerial images.

To this end, automated image understanding through Machine Learning (ML) and Computer Vision (CV) provides exciting potential for accelerating applications that rely on large-scale datasets, such as image-based aerial animal censuses. Employing machines to detect animals could greatly reduce the efforts required by humans, and therefore lead to vastly increased efficiency in the census process overall.

This thesis aims at advancing wildlife conservation efforts by means of automated machine vision methodologies. In a first step, this entails finding new ways to optimize CV algorithms for the task of animal detection in UAV imagery. In a second step, it requires procedures to reuse such detection models for new image data in the context of census repetitions for population monitoring. However, the benefit of machine vision reaches beyond a mere automation of photo-interpretation: a recurrent key principle of this thesis is the concept of interactivity, where CV models and humans work hand-in-hand by reinforcing each other. The result is a census monitoring environment for UAV images, in which machine vision technology actively assists humans in the process. Effectively, when all methodologies proposed throughout this thesis are combined, human annotation efforts are reduced to a fraction, and further simplified in complexity.

Chapter 2 addresses the challenges of employing state-of-the-art CV models, known as Convolutional Neural Networks (CNNs), for aerial wildlife detection. Multiple heuristics are presented to train such models properly, all of which target different obstacles of the model training process. Experiments show a significant increase in animal prediction quality, if a CNN is optimized in an appropriate way.

Chapter 3 employs this CNN for reusage over new data acquisitions, *e.g.* in a census monitoring setting. Simply running the CNN over a new dataset to predict animals directly is often not possible, due to differences in characteristics between the datasets known as domain shifts. This chapter presents methodologies to adapt CNNs to new datasets with minimal effort possible, and employs humans in the process in an interactive manner to do so. Results show that less than half a percent of the images need to be reviewed by humans to find more than 80% of the animals in the new campaign.

In Chapter 4, human annotation efforts themselves are addressed and reduced in complexity. Traditional settings require human annotators to draw bounding boxes around animals, which may become prohibitively expensive for large image datasets. This chapter instead explores the concept of weakly-supervised object detection, where only simple presence/absence information of animals per image is requested from the annotators. Unlike bounding boxes, an image-wide annotation can be provided in a second. It was found that a CNN, trained on this simpler information alone, is already able to localize animals by itself to a certain degree. However, if spatial bounding boxes are added for just three training images, the CNN predicts animals with the same accuracy as its fully-supervised sibling from Chapter 2.

Finally, Chapter 5 combines all findings and models into an integrated census software environment, denoted as *Annotation Interface that Does Everything* (AIDE). To the best of the author's knowledge, AIDE is the first software solution that explicitly integrates machine vision technology into the labeling process in an interactive manner: in AIDE, CNNs are used to predict animals in a large set of unlabeled data, and further learn directly from annotations provided by humans on the images. The result is a positive feedback loop where humans and machine reinforce each other. A conducted user study shows that machine vision support provides a four-fold increase in the number of animals found in a given time, compared to an unassisted annotation setting on the same dataset. At the time of writing, AIDE is actively employed by conservation agencies in Tanzania and under consideration by other forces around the globe for potential usage.

This thesis highlights the importance of interactive machine vision for wildlife conservation, and provides solutions that not only advance the field in a scientific context, but also have a direct impact on wildlife conservation through population monitoring.

Zusammenfassung

Der Verlust bedrohter Tierarten hat jüngst Werte erreicht, die einem sechsten Massensterben gleichkommen. Insbesondere Wirbel- und Säugetiere, wie etwa die Gattung der Nashörner, sind akut vom Aussterben bedroht und werden im Verlaufe folgender Jahre möglicherweise nicht mehr in freier Wildbahn anzutreffen sein. Handlungsbedarf zum Tierschutz besteht ungemein, doch müssen Erhebungen momentaner Tierbestände – etwa durch Populationszählungen – durchgeführt werden, bevor Massnahmen ergriffen werden können. In Anbetracht langzeitlicher Beobachtung von Tierbeständen besteht ferner Bedarf an periodisch wiederholten Zählungen.

Die neusten Entwicklungen in Forschung und Technik ermöglichen heutzutage Zählungen unvorhergesehener Genauigkeit und Dimension. Insbesondere der Einsatz unbemannter Luftfahrzeuge, umgangssprachlich bekannt als Drohnen, hat die Datenerhebung diesbezüglich signifikant vorangetrieben. Drohnen erlauben eine rasche und sichere Abdeckung von Wildtierparks durch Luftbilder und bilden dadurch eine zuverlässige Grundlage zur Bestandesaufnahme von Tieren. Den Hauptanteil an Arbeit bildet dabei jedoch nicht die Datenerhebung an sich, sondern die nachfolgende Tieridentifikation durch manuelle Fotointerpretation, welche aufgrund der grossen Datenmenge (i.d.R. mehrere zehntausend Luftbilder pro Datenerhebung) äusserst aufwändig ausfällt.

Eine Möglichkeit zur Vereinfachung der Tieridentifikation ist der Einsatz automatischer Bildinterpretation durch maschinelles Lernen. Computer-Algorithmen zur automatischen Tierdetektion bieten dabei die Möglichkeit, grosse Datensätze zu prozessieren und damit den manuellen Arbeitsaufwand für Menschen erheblich vereinfachen zu können.

Die vorliegende Arbeit beschäftigt sich entsprechend mit der Anwendung maschinellen Lernens auf das Thema des Wildtierschutzes. Ein erster Aspekt behandelt in diesem Zusammenhang neue Optimierungsmöglichkeiten automatischer Tierdetektionsalgorithmen auf Basis von Drohnenbildern. Das resultierende Detektionsmodell wird in einem zweiten Schritt weiterentwickelt, um mit minimalem Aufwand auf neue Datensätze aus wiederholten Erhebungen angewandt werden zu können. Das Potential maschinellen Lernens reicht dabei allerdings über eine einfache, passive Anwendung auf Datensätze hinaus – ein wiederholt anzutreffendes Prinzip ist zum Beispiel der interaktive Aspekt, in welchem Mensch und Maschine sich gegenseitig unterstützen und positiv beeinflussen, statt

unabhängig voneinander zu agieren. Das Produkt der Erhebungen dieser Arbeit bildet eine komplette Umgebung zur Tierzählung in Drohnenbildern, in welcher Detektionsalgorithmen die Nutzer aktiv im Interpretationsprozess unterstützen. Wenn alle propagierten Methoden kombiniert werden, kann der benötigte Arbeitsaufwand der Nutzer nicht nur auf einen Bruchteil der ursprünglichen Menge reduziert, sondern auch erheblich in der Komplexität vereinfacht werden.

Kapitel 2 bildet die Grundlage zum Training modernster Algorithmen zur Objekterkennung aus der Familie neuronaler Netze, angewandt auf Tierdetektion in Drohnenbildern. Dies erfordert Strategien, die verschiedenste Aspekte des Trainingsprozesses optimieren. Experimente zeigen eine signifikante Verbesserung der Detektionsqualität der Algorithmen, sofern adäquat trainiert, sowie prinzipielle Eignung der Methoden auf das Hauptanwendungsgebiet auf Basis drohnenbasierter Bilddatensätze.

Diese Detektionsmethoden werden in Kapitel 3 erweitert, um dem Szenario wiederholter Tierzählungen gerecht zu werden. Aufgrund unterschiedlicher Charakteristiken von Datensätzen ungleicher Jahre, beispielsweise durch Veränderungen in der Phänologie, können Detektionsalgorithmen nicht ohne Anpassung für neue Datensätze wiederverwendet werden. In diesem Kapitel werden entsprechend Methoden präsentiert, die die Ausführung der benötigten Anpassung mit geringstmöglichem Aufwand ermöglichen. Zu diesem Zweck wird unter anderem in einem interaktiven Kontext auf Eingaben von Nutzern zurückgegriffen. Das Resultat ist ein System, welches mehr als 80% der Tiere in neuen Datensätzen findet und dabei Nutzereingaben für weniger als 0.5% der Bilder benötigt.

Kapitel 4 adressiert die Nutzereingaben selbst und stellt Methoden zur Vereinfachung der Komplexität derer vor. Konventionelle Fotointerpretation erfordert die Markierung sämtlicher Tiere, beispielsweise durch Rechtecke. Diese Art der Eingabe birgt die Gefahr grossen Aufwandes, insbesondere für eine grosse Anzahl an Bildern. Dieses Kapitel untersucht stattdessen Möglichkeiten und Potential weniger komplexer Bildannotationen: anstelle von Rechtecken wird beispielsweise lediglich die Information benötigt, ob ein Bild eines oder mehrere Individuen einer Tierart enthält oder nicht. Im Vergleich zu räumlich definierten Annotationen kann eine bildweite Angabe der Präsenz/Absenz einer Tierart in Sekunden bewältigt werden. Resultate zeigen, dass Detektionsalgorithmen Tierindividuen erstaunlich präzise erkennen und sogar räumlich verordnen können, selbst wenn sie lediglich mit genereller Präsenz/Absenz von Tierarten trainiert wurden. Die Zugabe von räumlichen Koordinaten für lediglich drei Luftbilder führt darüber hinaus zu gleichwertiger Detektionsqualität solcher Modelle wie diejenigen aus Kapitel 2, welche mit erheblich aufwändigeren Annotationen trainiert wurden.

Kapitel 5 kombiniert schliesslich alle vorgeschlagenen Methoden und Prinzipien in einer Softwareumgebung namens "*Annotation Interface that Does Everything*" (AIDE), die unter anderem für Tierzählungen und Zählungsrepetitionen in Drohnenbildern geschaf-

fen wurde. AIDE ist womöglich die erste Software-Suite, die Algorithmen maschinellen Lernens direkt und interaktiv in den Fotointerpretationsprozess einbringt. Die vorgestellten Algorithmen werden in AIDE dabei nicht nur zur Detektion von Tieren verwendet, sondern lernen wiederum von den Eingaben und Korrekturen der Nutzer, um etwaige Fehler in weiteren Iterationen vermeiden und Vorhersagen auf Nutzerbedürfnisse optimieren zu können. Das Resultat dieses zyklischen Lernprozesses ist eine gegenseitige Stimulierung von Mensch und Maschine und schlussendlich eine deutliche Beschleunigung des Tierzählungsvorganges: Experimente zeigen eine Vervierfachung der Anzahl gefundener Tiere durch AIDE, wenn Nutzer durch Algorithmen unterstützt werden. AIDE wird momentan aktiv von Umweltschützern in Tanzania eingesetzt und steht ferner unter Evaluation durch weitere Organisationen.

Insgesamt erläutert diese Arbeit somit die Relevanz interaktiver, durch maschinelles Lernen unterstützter Ansätze für langfristige Wildtierzählungen. Sie bietet dabei eine Reihe an Lösungen, die nicht nur einen Beitrag zur theoretischen Wissensakkumulation in der Forschung beisteuern, sondern darüber hinaus auch einen direkten Einfluss auf den Tierschutz selbst haben.

Contents

	Page
Summary	v
Zusammenfassung	vii
Contents	xi
Acronyms	xiii
Chapter 1 Introduction	1
Chapter 2 Detecting Mammals in UAV Images	19
Chapter 3 Half a Percent of Labels is Enough	51
Chapter 4 Weakly-supervised wildlife detection in UAV images	71
Chapter 5 AIDE: AI for Image-based Ecological Surveys	87
Chapter 6 Synthesis	103
References	119
Acknowledgements	133
About the author	135
PE&RC Training and Education Statement	137

Acronyms

AI Artificial Intelligence

AIDE Annotation Interface that Does Everything

AL Active Learning

BoVW Bag of Visual Words

CNN Convolutional Neural Network

CV Computer Vision

DA Domain Adaptation

DL Deep Learning

HOG Histogram of Oriented Gradients

MAE Mean Absolute Error

ML Machine Learning

MLP Multi-Layer Perceptron

MSE Mean Squared Error

NMS Non-Maximum Suppression

OT Optimal Transport

ReLU Rectified Linear Unit

RF Random Forest

SIFT Scale-Invariant Feature Transform

SVM Support Vector Machine

TS Transfer Sampling

UAV Unmanned Aerial Vehicle

WSOD Weakly-Supervised Object Detection

Chapter 1

Introduction

1.1 Context

Species loss, in particular among vertebrates and mammals, is one of the most pressing issues in today's environment (Ceballos et al., 2015). Figures report population size reductions to critically low levels for a number of key species, such as the black and white rhinoceros (*Diceros bicornis* and *Ceratotherium simum simum*; Ferreira et al. (2015)) and African elephants (*Loxodonta*; Wasser and Gobush (2019)). Reasons for decline can be traced back to habitat loss (Kideghesho, 2009), as well as poaching and trade of ivory and horns (Biggs et al., 2013; Schlossberg et al., 2019). Actions to stem losses range from trade bans (Harvey et al., 2017) and armed patrols (Witter and Satterfield, 2019) to species re-introductions (Ferreira and Greaver, 2016). However, before measures for conservation can even be considered, animal *censuses* have to be carried out to determine the current vitality of populations. Moreover, with species, their habitat, and threats forming a dynamic system of interactions, census *repetitions* are required to *monitor* populations over time and predict disruptions in the long term (Collen et al., 2011).

Wildlife censuses are traditionally based on sample acquisitions, followed by extrapolations (Lewis, 1970). Common data acquisition methods include camera traps (Silveira et al., 2003), GPS collars (Blake et al., 2001), foot surveys (Jachmann, 1991), and manned aircrafts (Bayliss and Yeomans, 1989; Norton-Griffiths, 1978). All of these methods entail numerous limitations: camera traps and collars induce extrapolation and setup uncertainties (Meek et al., 2015), manned survey results suffer from observer bias (Caughley, 1974), and low-flying helicopters may severely disturb wildlife (Côté, 1996). Even more critically, employing helicopters is not only expensive, but dangerous, accounting for the majority of fatal injuries among biologists (Sasse, 2003).

Since recently, surveys are increasingly carried out using drones, also known as Unmanned Aerial Vehicles (UAVs). UAVs are lightweight aircraft that can be equipped with payloads like RGB or thermal infrared cameras. They work by flying over the area of interest while capturing high-resolution imagery or videos, in which any animal visible can be located and annotated with its position, or extent by means of a bounding box. On paper, UAVs forgo most of the limitations of manual surveys: they can be remotely controlled, which provides high operator safety, they are inexpensive, and their ability to acquire imaging data over large areas has the potential to significantly reduce human observer biases. Applications to animal censuses have thus been carried out and provided highly promising results (Hodgson et al., 2013; Linchant et al., 2015; Hodgson et al., 2018). However, UAVs pose one major challenge: in order to provide sufficient coverage of study areas, they have to acquire thousands to hundreds of thousands of images, each with very high (*i.e.*, sub-decimeter) resolution. This shifts the main workload to the stage of photo-interpretation which, if done manually, is error-prone, tiring, and expensive (Hollings et al., 2018).

1.2 Computer Vision for Animal Localization

One possible answer to the tedium of photo-interpretation is to automate the work using computers. Computers are fast and able to process large quantities of data with constant results, which is helpful when datasets consist of thousands of images. Teaching computers how to *e.g.* localize animals can be done through the help of Computer Vision (CV), which attempts to bring computers to understand higher-level semantic concepts in imagery. CV is typically combined with Machine Learning (ML), the process of making computers learn patterns in data by means of examples (Bishop, 2006). An ML algorithm, often called a *model*, is generally designed to provide predictions for one or more types of tasks. For example, in the case of animal localization, an appropriate task would be what is known in a more general CV context as "object detection."

Most common object detection models are trained in a *supervised* way, where the model is both given a set of images and so-called "ground truth" to learn from during training. This ground truth contains rectangular bounding boxes that indicate *where* in the image the object(s) of interest are present. Optionally, detectors may also perform classification, where every bounding box is also annotated with a label from a pre-defined set of classes. In this case, the model not only has to perform object localization in an image, but also has to tell *what* type of object it is. For animal detection, the set of label classes could for example contain animal species that roam in a given study area. Once a sufficient amount of ground truth has been collected, the model is trained to accurately recognize the objects in the training images and to be able to find the same type of objects in similar images beyond the training set.

1.2.1 Multi-stage object detection

Traditional ML models for object detection consist of multiple stages (Figure 1.1):

1. Identify *object proposals*, *i.e.* rectangular regions inside an image that are likely to contain some sort of object (Uijlings et al., 2013; Zitnick and Dollár, 2014). Not every proposal contains an object of interest (*i.e.*, an animal), but the idea is to obtain initial positions and provide bounding boxes for as many animals as possible in the image.

2. Calculate *feature descriptors* for every object proposal: features are supposed to capture essential appearance cues of an object and to best separate objects from background. For example, a descriptor activating strongly if a striped pattern occurs in an image could be useful to identify zebras. Example features include Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005), Scale-Invariant Feature Transform (SIFT) descriptors (Lowe, 1999), and Bag of Visual Words (BoVW) (Sivic and Zisserman, 2003).

3. Employ a classifier, such as a Random Forest (RF) (Breiman, 2001), that receives said feature descriptors and predicts a label accordingly (*e.g.* the animal species, or else "background", if the respective object proposal does not contain an animal). The final output is a subset of proposals that have been predicted as animals.

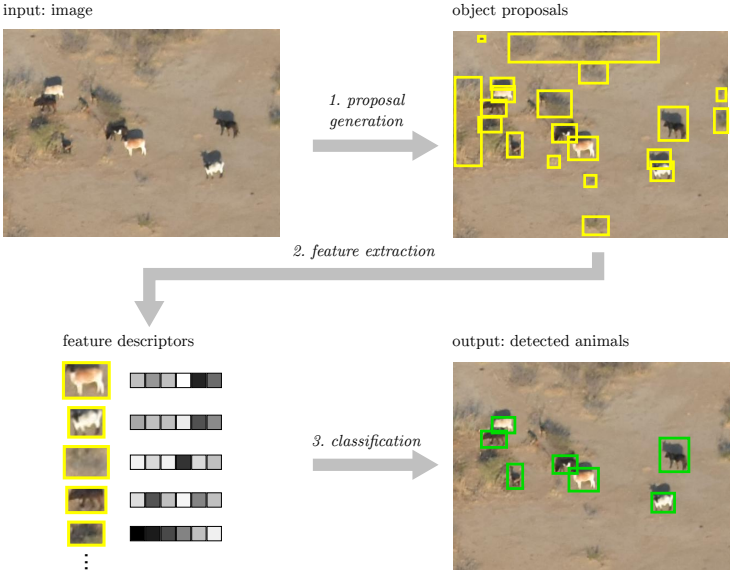


Figure 1.1: Traditional object detectors consist of multiple stages: proposal generation (yellow; top right), extraction of features (bottom left), and classification of each proposal into animals (green; bottom right) or background.

Multi-stage detectors had found common use in various CV applications, including aerial animal detection (Rey et al., 2017). However, their intrinsic mechanics entail some crucial limitations: on the one hand, these detectors are slow, since thousands of object proposals have to be evaluated per image in order to ensure that as many animals as possible can be detected. On the other hand, their manually engineered set of feature descriptors might not be discriminative enough for the problem and the images at hand. For example, although the aforementioned striping pattern detector conceptually makes sense for identifying zebras, it most likely will not work in reality: animals may adopt a pose that obscures their striped torso; or they may be oriented in an unexpected way the pattern matching process was not designed for, an effect that is particularly strong in aerial points of view.

1.2.2 Using deep learning for detection

For the aforementioned and other reasons, multi-stage object detectors have these days mostly been phased out by Deep Learning (DL) (LeCun et al., 2015). In CV, the most commonly used DL models are in the form of Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012). Due to their superior performance, CNNs are these days used across many different ML tasks, and employed in devices from smartphones to self-driving cars. The primary distinction of DL compared to more traditional ML models is that they do not rely on hand-crafted feature descriptors, but jointly learn a classifier *and* a series of features, a process known as *end-to-end* training. The main advantage behind end-to-end training is that features can be specifically optimized for the task at hand. Features in CNNs are calculated by means of convolutions, which involves applying a dot product between a (learnable) filter matrix and a region around every position in the input. These convolutions are typically grouped into *layers* and applied sequentially, which results in the set of features of one layer being dependent on all the features of the previous one(s).

Figure 1.2 shows example features produced by a CNN ("feature tensors"). In this simplified diagram, the CNN employs three *convolution* layers, each producing one of the three output tensors shown (top right). Every layer contains a number of filters that are convolved over the previous layer's output, respectively the input image for the first layer. In CNNs, filters are spatial (*e.g.* of size 3×3). Since each layer applies its filters over already convolved features, layers higher up the hierarchy are able to capture increasingly high-level semantic concepts. For example, the initial layers, such as the first one that receives the bare image pixels as an input, typically encode small, geometric properties (*e.g.* edges), and layers up the hierarchy capture more abstract concepts like animal body parts, entire animals, and more. However, these latter concepts are typically also larger in size compared to geometric edges and patterns, which requires filters in later layers to cover a larger spatial extent. In CNNs, this is achieved through spatial downsampling. In addition to convolution layers, CNNs often employ *pooling* layers that summarize the contents of feature tensors in a given spatial extent. Rather than convolving a feature tensor with a filter, pooling layers compact the tensors in a non-learnable way, *e.g.* by reducing tensor values in a neighborhood to their average, or maximum value. This results in a reduction in resolution that is required for the larger-scale parts layers higher up the hierarchy are supposed to capture.

The transition from geometric to semantic concepts can also be seen in Figure 1.2, where the first set of features focuses on edges between bright and dark pixels, while later features highlight animal positions. Employing more layers and/or more filters per layer increases the CNN's capacity to learn a wider variation of concepts, such as different animal species, orientations, and the like. A final layer then calculates a weighted combination of all features over a spatial position and yields a value that indicates how likely the position

is to contain an animal or not (Figure 1.2, "probability map"), thereby providing spatial positions of identified animals. The final model is often more complex than just a sequence of convolutions and pooling layers, but these can be seen as the main building blocks of CNNs. For a more in-depth explanation of DL models, the reader is advised to consult dedicated works, such as Goodfellow et al. (2016).

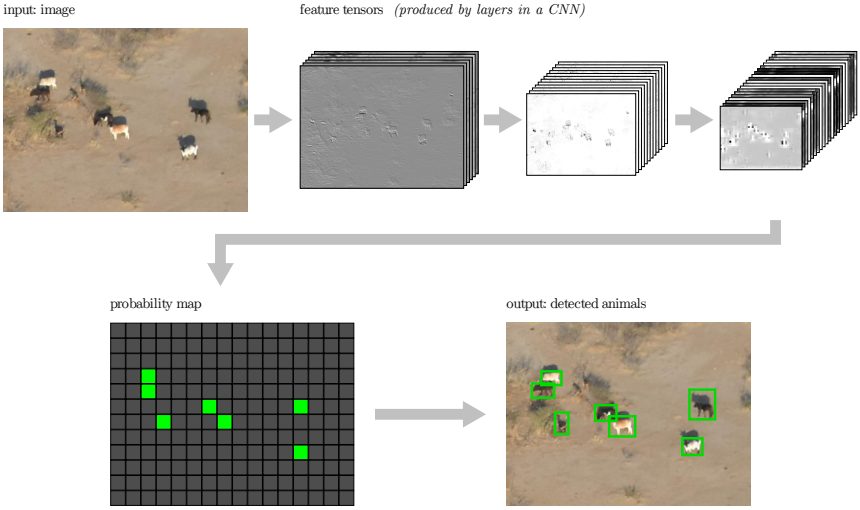


Figure 1.2: A simple CNN architecture for object detection. Many of these models forgo the multi-stage approach (*i.e.*, object proposals followed by feature extraction and classification). Instead, they predict spatial feature tensors, and eventually animal locations and extents, directly.

The ability of CNNs to learn task-specific features has led to significant accuracy increases for various CV tasks, including object detection. Example object detector models proposed over the years include Faster R-CNN and its predecessors (Girshick et al., 2014; Girshick, 2015; Ren et al., 2015), SSD (Liu et al., 2016), YOLO and evolutions (Redmon et al., 2016; Redmon and Farhadi, 2017), and RetinaNet (Lin et al., 2017), with all but the first family being variants of single-stage models that predict locations and bounding boxes directly. This property, together with advancements in computational hardware, makes single-stage CNN detectors extremely fast, which is vital for the dataset sizes of UAV acquisitions.

CNN-based models are predominantly trained on, and optimized for, classical CV datasets like ImageNet (Deng et al., 2009; Russakovsky et al., 2015), which are primarily composed of *natural* images taken from a ground-level perspective. In such a setting, objects often cover a large portion of the images and exhibit highly discriminative variations of patterns

and geometrical shapes. This allows CV models to detect objects very reliably. In the case of aerial imagery, however, these properties do not hold. Instead, objects are significantly smaller, and differences in appearance between object classes may be more subtle. The challenges of overhead imagery have been addressed in the field of remote sensing (Cheng and Han, 2016), to which aerial animal localization can be counted as well. Although a variety of different strategies have been proposed for remote sensing applications, a reoccurring observation that can be made is that models like CNNs primarily have to learn the higher-level features in a different way. Lower-level concepts like edges and geometric patterns in turn remain the same (Figure 1.3). This basically means that conventional CV object detectors can still be used for remote sensing tasks, although with adaptations in model design and training.

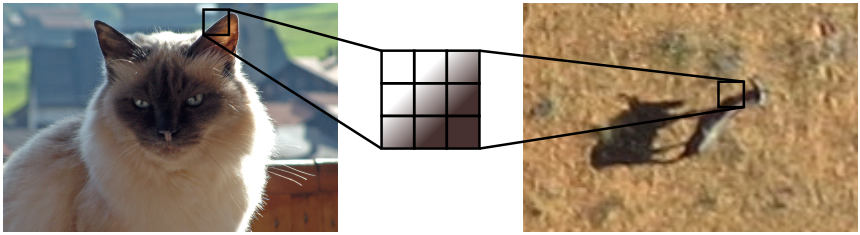


Figure 1.3: Objects have different appearances in natural images (left) compared to overhead imagery (right), but lower-level features like edges (middle) can be found in both.

The task of animal localization poses one additional challenge in the form of object *scarcity*. As an example, Figure 1.4 shows a full UAV image acquired over the Kuzikus wildlife reserve in Namibia[1] and is part of one of the datasets that will be used throughout this thesis. The image contains two mammals, highlighted and enlarged in blue. As can be seen, the area covered by them is minuscule, especially compared to the vast surrounding background. If these two animals are already challenging enough for humans to spot, an out-of-the-box CV object detector is bound to miss them completely: animals are too small to provide sufficient appearance context for discrimination, and their number is too low to form a large enough part of the data distribution (*i.e.*, they are underrepresented with respect to the highly abundant background areas). Even worse, the scarcity of animals means that a large portion of UAV images are bound to be "empty," *i.e.* they do not contain any animals at all. This causes the dominance of background areas over animals to be even greater. As a result, conventional detection CNNs are likely going to miss the animals, simply because they are overwhelmed with the sheer number of background pixels. Hence, new methodologies are required to properly employ CV models, in particular CNN-based object detectors, to the task of identifying animals in UAV images for censuses.

[1] `https://kuzikus-namibia.de/xe_index.html`

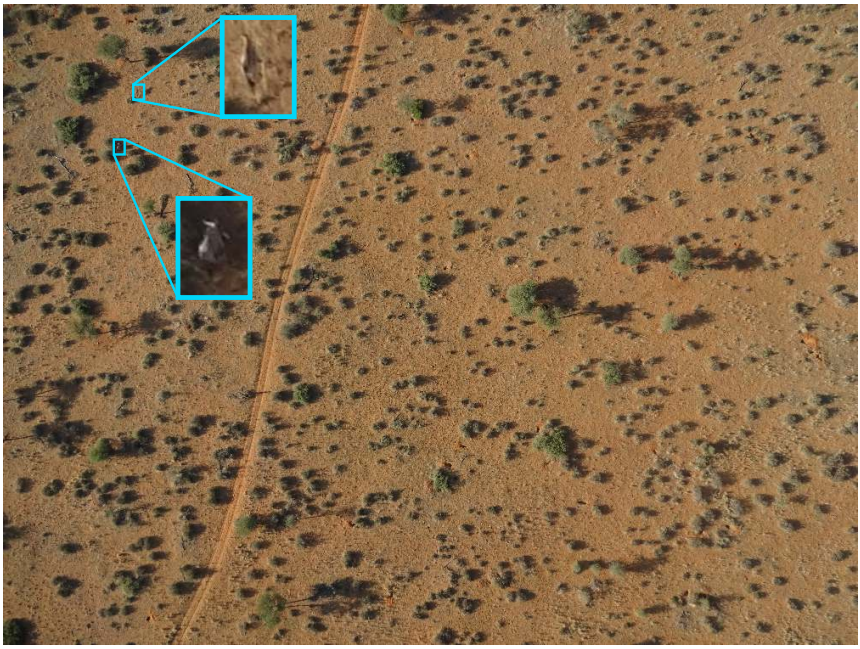


Figure 1.4: UAV images pose new challenges to CV object detection models, since their distant viewpoint causes objects of interest like animals (blue; enlarged) to become minuscule in area covered. An out-of-the-box CV object detector would likely not be able to identify the two animals present in the image shown.

1.3 Domain Shifts between Image Campaigns

Until this point, elaborations have been concerned with applying CNN detectors on a single aerial image dataset only. This may apply for an instantaneous survey of an animal population, where every image is acquired in a single image campaign. However, it falls short for population monitoring, where census *repetitions*, and therefore multiple campaigns at different points in time, need to be carried out. In such a scenario, the challenges multiply.

One initial attempt to tackle repeated censuses could be to train a detection CNN on the images from the first acquisition campaign, and re-use it for prediction on all following ones. Unfortunately, this is suboptimal for reasons explained below.

Consider Figure 1.5: shown here are parts of two different UAV images, again with animals highlighted and enlarged in blue. Both images have been acquired in the Kuzikus wildlife reserve, albeit over different parts of the park, as well as in different years. As can be seen, they have a fairly different appearance from each other. For humans, it is comparably possible to spot the animals in both images, perhaps with a bit of training. To machines, however, this does not apply—for example, if a CNN has been trained on the first set of images (left), it will likely miss the animals in the second set (right), and vice versa, as it has never been exposed to images with such different appearance.

Figure 1.5: Images from different UAV campaigns may exhibit inherent discrepancies, like differences in soil composition, animal types, and the like. These are known as domain shifts and often cause CV models to fail, unless said models are adapted to these new situations.

This problem is known in ML as *domain shift*: the two images are part of different datasets (*domains*), which feature significant variations (*shifts*) between each other. For aerial imagery, these shifts can be decomposed into a number of contributing factors, such as:

- *Study area:* images may be captured over different landscapes featuring different soil types.
- *Phenology:* one dataset may be collected in summer, and the other during winter, with the vegetation being at a different stage each time.
- *Hardware:* shifts can also occur if the UAV is operated at a different height above ground, with a different speed (*cf*. blurriness of images), or with different cameras.
- *Animal density and species:* finally, discrepancies can also occur on the number of animals in the datasets—an ML model generally responds differently if the ratio between animal and background pixels changes. Likewise, a model trained to *e.g.* identify zebras may not automatically be able to recognize rhinos.

Procedures to overcome domain shifts belong to the family of *Domain Adaptation* (DA) methods (Patel et al., 2015; Tuia et al., 2016). These methods assume the presence of a so-called *source* domain, an ML model trained on it, and a *target* domain that is different from the source. Their goal is to make the source model work reliably on the target

dataset by using no or as little ground truth as possible from the target domain. For example, if the images in two domains were captured with different types of cameras, DA methods could be employed to make the detector recognize animals even with different radiometric signatures, *e.g.* by finding detections in the target domain that are most similar to animals in the source, and assimilating their predicted feature descriptors to similar values accordingly.

Over time, a number of DA methods have been proposed, some of which have also been tested in remote sensing settings (Tuia et al., 2016; Tuia and Camps-Valls, 2016). For example, a mode followed by many DA methods is to encourage a model to predict features for the source and target domains that are as similar as possible. The rationale behind this is that with aligned features, the model should be able to perform its task well independently of the data domain at hand. Most strategies attempt to do so in an *unsupervised* way, which in a DA context means that no ground truth for the target domain is needed[2]. Unfortunately, unsupervised feature alignment does not work well in the highly imbalanced setting posed by aerial animal localization. With animals appearing in low numbers, there is a too high risk of unsupervised DA methods to assimilate animal with background features, which results in the model not recognizing animals at all anymore. Since there is no ground truth for the target domain, this false alignment between animals and background cannot reliably be prevented. *Semi-supervised* DA strategies exist that include a small amount of target ground truth. This might prevent the problem of harmful source-to-target class associations, but it cannot solve the fundamental class imbalance problem. Doing so requires DA methodologies that are able to cope with class imbalance from the ground up, which has not yet been explored for the application of scarce animal detection.

1.4 Including Humans in the Process

A final theme of this thesis addresses the role of humans[3] in the process. Besides being beneficiaries of animal detections, humans extensively contribute by providing image annotations (ground truth), needed to train the models. Especially CNNs are known to require vast amounts of ground truth to work properly. For example, the fundamental detector CNN employed throughout this thesis was trained with 450 full-sized UAV images, annotated with bounding boxes around animals. This constitutes 60% of the whole dataset described in Chapter 2.2.1. Although an explicit study on the minimally required number of training images was not conducted, observations have shown model performance to suffer critically below a few hundred training examples. Moreover,

[2]Ground truth is still required for the source domain in order to train the initial CNN.

[3]For wildlife conservation, this includes *e.g.* park rangers, authorities, and ecologists, but not necessarily algorithm developers.

even if models could be trained with less data, creating the training set would still be an involved process: the animals have to be identified and annotated, and the images themselves need to be retrieved from the large number of images that do not contain an animal. In sum, this means that a plain detector CNN might be able to reduce the required photo-interpretation work a bit, but not to a satisfactory degree.

1.4.1 Facilitating User Inputs

There are several conceivable incentives to further lower the workload for humans. A first one addresses the *complexity* of the user input. For animal censuses, user input primarily comprises bounding boxes around animals. The task of drawing bounding boxes itself can be challenging and has found to be a source of uncertainty in conventional CV problems (Papadopoulos et al., 2017a). In aerial imagery, it is additionally cumbersome due to the minuscule size of animals: rectangles have to be drawn tightly, which requires repeatedly zooming in. Sometimes, it may become hard to distinguish animals from the shadow they cast, depending on the illumination conditions. Finally, animals are often found in herds or flocks, making the annotation task imbalanced: the majority of images may be free of animals, but if they do, they frequently contain them by significant numbers.

Ideally, the easiest way to facilitate the labeling process would be to simply drop the necessity of drawing bounding boxes, but have humans merely state whether an image contains animals, or not. This forfeits the applicability of traditional object detectors, since they precisely need spatial positions in order to learn their task. In this context, however, it is important to note that the spatial position itself is only supposed to play a secondary role for animal detectors: in theory, where in the image an animal comes to lie should not have an influence on whether it can be detected; rather, it is the animal's *appearance* that is supposed to be the main determinant. Hence, an animal detector should in theory be able to automatically identify animals based on reoccurring appearance cues, if it is being told *whether* an image contains animals or not.

As it turns out, this concept has been studied in CV and is known as Weakly-Supervised Object Detection (WSOD). In spirit, these models are still object detectors that yield spatial predictions, but the key distinction is that they can be trained using *weak* ground truth, *i.e.* labels of lower complexity. Image-wide presence or absence of species is a variant of a ground truth that is weaker than spatial positions or bounding boxes, as it provides less detailed information. If such models could be employed for animal censuses, they could greatly reduce the time spent by humans on annotating each image. For example, instead of spending several minutes on drawing bounding boxes around the ostriches in Figure 1.6, one simple label ("this image contains animals") would suffice.

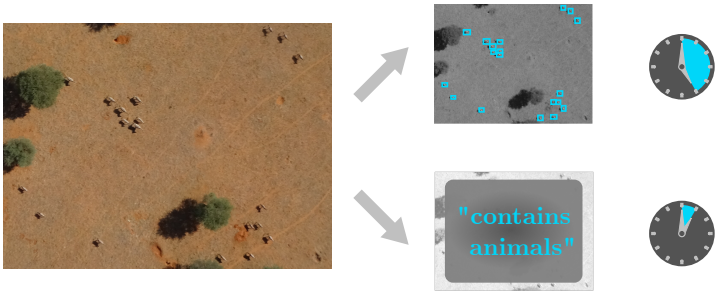


Figure 1.6: Drawing bounding boxes around each animal can be a tedious process for humans (top right). Instead, simply stating whether an image contains one (or more) animals or not, without specifying where, could reduce efforts and time significantly (bottom right).

WSOD models applied to CV problems have shown to reach a fair performance, at the advantage of requiring significantly simpler and less costly ground truth (Oquab et al., 2015; Bilen et al., 2015; Bazzani et al., 2016). Some of the methods exploit similarities of label class objects across images to determine where in the image the respective object was identified; others resort to the spatial output activations of CNNs to do so. Irrespective of the method used, WSOD could greatly alleviate efforts and resources spent on the labeling input, if found to be applicable to aerial animal censuses.

1.4.2 Active Learning for Intelligent Training Sample Selection

A second approach that has the potential to reduce the human workload draws on the concept of Active Learning (AL). AL is a strategy that limits the amount of training data to a subset of samples in such a way that the model is still capable of learning its task, perhaps even with a better performance (Settles, 2012). The underlying hypothesis behind AL is that training data may exhibit some degree of redundancy that can be reduced. For example, UAV images may repeatedly show the same background compositions, or the same animals or species. If these images do not contain enough novel information, they do not contribute to model performance improvement, and can therefore be removed from the training set. Selecting only informative images has the upshot that less training data, and with that less user input to obtain ground truth labels, is required. The challenging part of AL is to identify *which* data points to choose—theoretical considerations, such as to remove images of similar appearance, might not hold to this end. Rather, AL infers the degree of usefulness from the *model*: for example, an image can be considered "useful" if it contains a lot of animals that the model failed to predict. Learning from this example could improve model performance more than exposing it to images where it already learned its task well. Choosing the appropriate images for training is at the core

of AL, and is referred to as the AL *criterion*. Another issue is that AL is applied straight from the start of the labeling stage, where the ground truth is not yet known. Hence, AL involves both the ML model (to determine the degree of usefulness of an image) *and* the annotators (to receive the labels for it) at the same time.

AL is particularly suitable for remote sensing problems, where a minimization of field work is crucial. AL criteria have thus been evaluated on remote sensing problems (Tuia et al., 2011a,b), but as with DA, the situation with respect to animal censuses and their class imbalance is unclear.

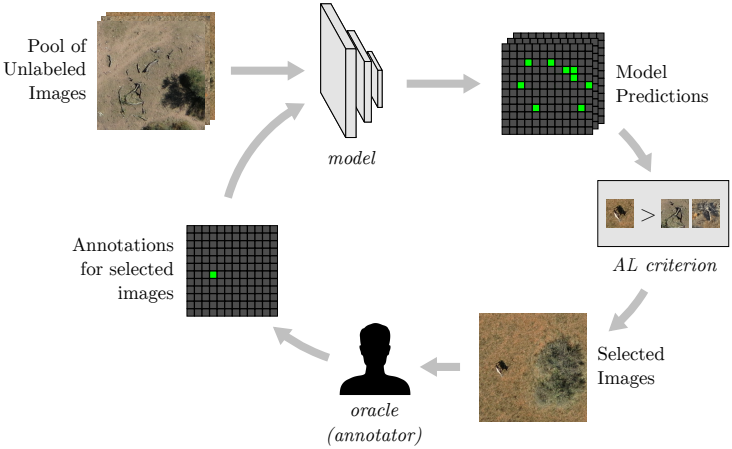


Figure 1.7: An AL loop begins with a pool of unlabeled images (top left), which are then predicted by the model (top). Since not all images contribute enough to model improvement, an AL criterion (right) selects the most appropriate ones. These are then labeled by the oracle, in this case an annotator (bottom). Those labels are then fed back into the model for training, and the loop starts anew.

The concept of a typical AL workflow is shown in Figure 1.7. As a precondition, AL assumes the presence of a model, a large set ("pool") of unlabeled images, and a so-called "oracle" that is able to provide ground truth for a limited number of images—in our case, this is the role of the human annotator(s). AL starts by running the model over the unlabeled images in order to obtain animal predictions. These predictions are then evaluated by the AL criterion, which ranks them with respect to their "usefulness" of the constituting data points. Many AL criteria tie the notion of usefulness to the prediction confidence of the model for a data point. For example, the Breaking Ties criterion (Luo et al., 2005) considers images most useful if the difference between the most and second-most confidently predicted class is low. The most highly ranked images are subsequently presented to the oracle, which provides ground truth for them accordingly. Finally, the

resulting image-label pairs are used to (re-) train the model, in the anticipation that the model would learn enough relevant concepts to better predict the animal locations next time. This completes one AL cycle, and the process starts anew, until the model performance is deemed satisfactory, or else a sufficient amount of the images has been labeled.

In this sketched scenario, AL essentially resembles a form of human-in-the-loop scheme, thereby making the machine vision process *interactive*. If successful, AL has the potential to significantly reduce the workload required. However, the significance of interactivity reaches further. As much as computers are capable of voluminous data processing, humans have the ability of high-level semantic reasoning. Machines need hundreds of training examples to be able to recognize an animal, while humans can do so with just one or two. CNNs may fail if exposed to unseen images, but humans generally have no issues in recognizing animals across domain shifts. Models do not know if they are accidentally aligning animals with background representations, but a human can prevent such mishaps through visual feedback. In turn, machines have the power to process large amounts of data and can filter them to only request user inputs for what is needed, precisely through the AL criterion. This again is likely to improve the quality of human inputs, since they can focus on fewer, but more relevant images, instead of clicking through loads of scenes that do not contribute to the model performance, let alone the objective of animal localization. In sum, these factors imply that interactive systems, with humans and a machine working hand-in-hand, have tremendous potential to be beneficial for *all* of the aforementioned challenges.

1.5 Research Gaps

With species loss and habitat diminution accelerating, the need for periodical animal censuses is more relevant than ever. Technology is increasingly being embraced to do so, thereby paving the way for potentially unprecedented census efforts. However, the employment of technology at this point in time predominantly ceases after the data acquisition stage. Essentially, machine vision assistance is not yet pursued to the required degree.

In consequence, the field is in need for investigations on the potential of ML and CV, and this based on multiple perspectives. From a CV perspective, making models work in the setting of high-resolution aerial imagery is highly challenging. Although a few works exist on animal detection from distant viewpoints (van Gemert et al., 2014; Yang et al., 2014; Fang et al., 2016; Xue et al., 2017), none of them addresses the scale and complexity of censuses in the wild. Ways of employing state-of-the-art CV models for animal detection thus need to be studied further. Such models are essential also for follow-up work, such as the mentioned challenges induced by domain shifts.

However, the identifiable gaps reach beyond conventional CV perspectives. Wildlife censuses establish a context that heavily involves humans as decision makers. As such, it is imperative to not only find ways to address their needs, but also to actively integrate them into the census process. This pivots the scope at multiple levels: on the one hand, concepts need to be investigated that reduce the *complexity* of human inputs needed, *e.g.* through WSOD. On the other, cutting down the *amount* of manual work, *e.g.* through AL, would likely prove advantageous, in particular for the ever-increasing data amounts to be expected from upcoming censuses.

Finally, solutions need to be found that convey these ideas to the target audience. Naturally, a large majority of the end-users of machine-assisted animal censuses are state park rangers, conservation agencies, and ecologists. This corroborates that machine vision for animal conservation is a multi-, perhaps even an interdisciplinary field of research. Crucially, it also means that methodologies might not easily be accessible to end-users as anticipated. With this in mind, proposed methodologies need to be made available to these end-users, for example through an interactive labeling interface that does not require expert knowledge in the fields of ML or CV.

1.6 Objectives

The main objective of this thesis is to establish an interactive animal census environment, with primary focus on population monitoring through aerial surveys. The amounts of data produced by the latter prohibit pure manual intervention and require a strong integration of automation. In turn, the complexity of the task likewise necessitates skilled user input into the process. Unifying both aspects into one streamlined workflow is still to be pursued. In order to do so, this thesis addresses the following research questions (RQs):

RQ 1 How can state-of-the-art DL models be trained for the task of aerial image-based animal localization?

RQ 2 What are the necessities and possibilities of CV in the context of animal population monitoring through census repetitions?

RQ 3 In which ways can the workload of human annotators in censuses and census repetitions be reduced, respectively optimized?

RQ 4 How can machine vision strategies and methodologies for animal censuses be conveyed to the target audience in an applicable manner?

1.7 Contributions

This thesis provides answers to the aforementioned questions in four chapters that interact as shown in Figure 1.8. In particular, they elucidate the following aspects:

Chapter 2 presents best practices for training CNN-based object detectors to recognize animals in UAV imagery (**RQ 1**). The resulting model significantly alleviates the workload for human operators (**RQ 3**). It further serves as a foundation for all subsequent chapters. This chapter is based on the following publication:

> **Kellenberger, B.**, Marcos, D., and Tuia, D. (2018c). Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153.

Chapter 3 introduces a new domain adaptation criterion that is specifically designed for repeated wildlife censuses (**RQ 2**). The proposed strategy resorts to AL and intends to accelerate and optimize label retrieval (**RQ 3**), and further has potential for inclusion in the ultimate target of an interactive census monitoring environment presented in Chapter 5. The chapter draws inspiration from the following work:

> Damodaran, B. B., **Kellenberger, B.**, Flamary, R., Tuia, D., and Courty, N. (2018). DeepJDOT: deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)(joint first author)*,

but is principally based on the following publication:

> **Kellenberger, B.**, Marcos, D., Lobry, S., and Tuia, D. (2019a). Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 57(12):9524–9533.

Chapter 4 addresses the feasibility of employing WSOD in the process of animal annotation. This provides insights into ways of facilitating the annotation procedure by reducing its complexity (**RQ 3**). The proposed model sees implementation in the environment that emerges from Chapter 5. This chapter is based on the following work:

> **Kellenberger, B.**, Marcos, D., and Tuia, D. (2019b). When a few clicks make all the difference: improving weakly-supervised wildlife detection in UAV images. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRw)*.

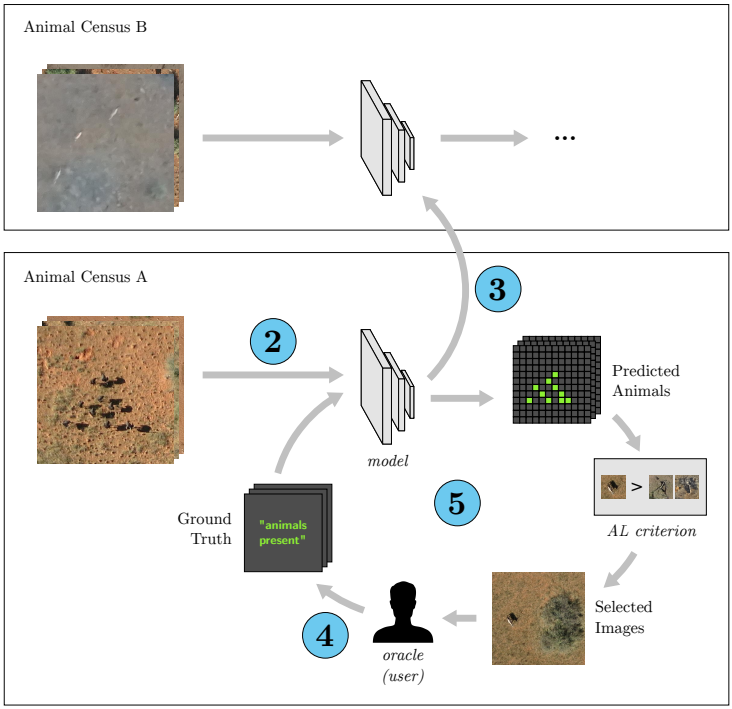


Figure 1.8: Overview of the interactive animal census monitoring framework proposed in this thesis. Numbers in blue refer to the main areas of contribution per chapter.

Finally, **Chapter 5** presents an annotation framework for animal labeling that integrates state-of-the-art machine vision tools into the labeling process while hiding technicalities from the user (**RQ 4**). The result is a labeling platform that employs CNN-based animal detectors (**RQ 1**) in an AL loop together with humans to empower and facilitate censuses as a whole (**RQ 3**). This chapter has been submitted as follows:

> **Kellenberger, B.**, Tuia, D., and Morris, D. (in revision). AIDE: accelerating image-based ecological surveys with artificial intelligence. *Methods in Ecology and Evolution*.

Chapter 2

Detecting Mammals in UAV Images: Best Practices to address a substantially Imbalanced Dataset with Deep Learning

This chapter is based on:

> **Kellenberger, B.**, Marcos, D., and Tuia, D. (2018c). Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153.

Abstract

Knowledge over the number of animals in large wildlife reserves is a vital necessity for park rangers in their efforts to protect endangered species. Manual animal censuses are dangerous and expensive, hence Unmanned Aerial Vehicles (UAVs) with consumer level digital cameras are becoming a popular alternative tool to estimate livestock. Several works have been proposed that semi-automatically process UAV images to detect animals, of which some employ Convolutional Neural Networks (CNNs), a recent family of deep learning algorithms. CNNs have proved very effective in object detection in large datasets from computer vision. However, the majority of works related to wildlife focuses only on small datasets (typically subsets of UAV campaigns), which might be detrimental when presented with the sheer scale of real study areas for large mammal census. Methods may yield thousands of false alarms in such cases. In this chapter, we study how to scale CNNs to large wildlife census tasks and present a number of recommendations to train a CNN on a large UAV dataset. We further introduce novel evaluation protocols that are tailored to censuses and model suitability for subsequent human verification of detections. Using our recommendations, we are able to train a CNN reducing the number of false positives by an order of magnitude compared to previous state-of-the-art. Setting the requirements at 90% recall, our CNN allows to reduce the amount of data required for manual verification by three times, thus making it possible for rangers to screen all the data acquired efficiently and to detect almost all animals in the reserve automatically.

2.1 Introduction

Livestock censuses play an important part in the ever-ongoing fight against the rapid decline of endangered large mammal species (Linchant et al., 2015). Knowing the exact number of individuals as well as their last known location sheds light on environmental requirements for different species (Gadiye et al., 2016), the developments of species reintroductions (Berger-Tal and Saltz, 2014), and can be of great help in anti-poaching efforts (Piel et al., 2015).

Identifying and counting animals in remote areas has traditionally been carried out manually, using methods such as surveys from manned aircrafts (Bayliss and Yeomans, 1989; Norton-Griffiths, 1978), camera traps (Silver et al., 2004) and other manual methods (Jachmann, 1991). For a long time, such campaigns were the only means of getting rough estimations of animal abundances, but they come with substantial flaws: *(i.)* they pose great risk on human operators who have to get close to armed poachers and wild animals; *(ii.)* they are expensive, requiring many man-hours of surveying, and *(iii.)* they might lead to accuracy deficiencies due to the limited extents that can be monitored. Camera traps have come down in cost (Nazir et al., 2017) but still require risky in-field installation and maintenance. Manned aircrafts overcome this problem, but are expensive and depend on human operators who might disagree and introduce estimation errors (Schlossberg et al., 2016; Bouché et al., 2012). Therefore, traditional methods generally lead to high monetary costs or raise safety concerns. These factors are particularly limiting in remote areas like the African savanna examined in this study.

A promising direction to address these issues is to employ UAVs for monitoring purposes (Hodgson et al., 2018). UAVs are remotely controlled, inexpensive aircrafts that can be equipped with sensing instruments such as point-and-shoot cameras. As such, UAVs alleviate both the risk on operators and the financial pressure on the data acquisition (Linchant et al., 2015). Furthermore, the bird's eye view allows reaching otherwise inaccessible areas from a safe distance. However, bypassing human counting inevitably requires more time to be spent on the analysis of the acquired data. Although the task of manual photo-interpretation does not expose the operator to the risks involved in field work, it can become prohibitively expensive for large UAV campaigns, which often generate tens of thousands of images. Works exist that employ experts to manually find animals in aerial images (Díaz-Delgado et al., 2017; Hodgson et al., 2013), but they were indeed limited to small study areas.

This problem can be alleviated by automatically selecting and showing to the operator only the images that are most likely to contain an animal, which typically account for a very small fraction of the total. Although this is bound to introduce some false negatives, one can leverage the improvements obtained in recent years in the computer vision field of object detection, which has already found a range of applications in remote sensing, to

minimize their impact. Our objective is to use machine-based object detection to train a model on a subset of the data that has been manually annotated and use it to predict the presence of objects in the rest of the dataset or in new acquisition campaigns. Object detectors work by extracting expressive image statistics (features) at locations of interest and using these features to classify each candidate location into a specific object class (*i.e.* "animal") or into a background class corresponding to the absence of any object of interest. Multiple approaches have been proposed to this end (Ren et al., 2015; Redmon et al., 2016; Dai et al., 2016), of which some have been applied to animal censuses. For example, several detectors can be found in the works of Chamoso et al. (2014) and van Gemert et al. (2014), both tackling the problem of cattle counting. Similarly, Andrew et al. (2017) deploy deep object detectors and trackers to detect cattle, but using very low-flying UAVs.
Recent works consider the detection of large mammals in the African savanna: Yang et al. (2014) and Xue et al. (2017) employ artificial neural networks on high resolution satellite imagery. Ofli et al. (2016) consider UAV data (the same data used in this study) and also provide a comprehensive set of annotations acquired by volunteers in a crowd-sourcing setting. In Rey et al. (2017), authors build on that dataset and propose a classification system based on an ensemble of classifiers, each one specializing in detecting a single animal. We will use their methodology as a baseline for our experimentations.

Despite the theoretical advantage of detectors to process large datasets quickly, many of such works have been showcased on relatively small and confined study sites (Hollings et al., 2018), as for example a set of images containing at least one animal, or a balanced dataset with equal occurrences of the animals and background classes. Even if this is acceptable for academic exercises, it has two consequences: *(i.)* the small scale, and therefore the probable dataset autocorrelation, does not allow to assess the performance of the method, were it to be upscaled to much larger areas; *(ii.)* the high concentration of individuals in the selected region hides the effect that covering vast, empty swaths of habitat has on the number of false positives that have then to be manually corrected. The few works that account for a realistic positive (animal) / negative (background) balance in a large mammal reserve (Rey et al., 2017; van Gemert et al., 2014) show that, at a recall rate of 80%, at least 20 false positives should be expected for each true positive. This can have a big impact on the effort required for manual verification, limiting the advantage of using an automatic detector in the first place. Our aim is to reduce the manual effort required by leveraging state-of-the-art object detection models and exploring a number of proposed training recommendations to adapt such models to this challenging and extremely unbalanced setting.

For object detection models, we consider recent developments in deep learning; more specifically, we deploy CNNs (Redmon et al., 2016). CNNs are nowadays the base building blocks of most computer vision pipelines and have also proven to be extremely successful in remote sensing (Zhu et al., 2017). Unlike traditional models, CNNs do not only train

Figure 2.1: Examples of animal-to-background confusion: dead tree trunks (2.1a) can easily be mistaken for ungulates (2.1b); shadows of dirt mounds (2.1c) may look like ostriches (2.1d).

a classifier, but include the feature extraction part in the training process, which yields features that are of particular validity for the problem at hand. This is highly valuable in the case of animal detection: animals show a substantial appearance heterogeneity due to their species, color and pose variations, and also due to external conditions such as motion blur, sensor differences, and different illumination settings. Furthermore, if seen from above, animals oftentimes tend to be hard to distinguish from various stationary objects like tree trunks, rocks, and dirt mounds (see Figure 2.1 for examples). Without end-to-end trained features, the result of this heterogeneity would increase the risk of automated systems to either miss animals because their appearance does not resemble what the model has learned (false negatives), or else falsely detect background objects that look like animals (false positives).

Summing up, we aim at taking a step towards solving the task of large mammal census in an African savanna wildlife reserve using UAV still imagery and deep learning. We present a series of recommendations that enable deep CNN models to address these issues. We showcase the recommendations on a dataset that features a realistic proportion between animals and background, which is challenging for detectors due to the small sample size and heavy class imbalance. Compared to the current baseline on our dataset (Rey et al., 2017), a CNN trained with these recommendations is able to reduce the number of falsely detected animals by an order of magnitude, while still being able to score a high recall (up to 90% of the animals present are detected). Moreover, since the (overall fewer) detections of our model are spread across a much lower number of UAV images, significantly less manual verification is required.

2.2 Data

2.2.1 Study area and ground truth

We base our studies on a dataset acquired over the Kuzikus wildlife reserve in eastern Namibia[1]. Kuzikus is a private-owned park located at the edge of the Kalahari desert and covers an approximate area of 103km^2. According to the park's estimations the number of large mammals in the park exceeds 3000 individuals and consists of more than 20 species, such as Greater Kudus (*Tragelaphus strepsiceros*), Gemsboks (*Oryx gazella*), Hartebeests (*Alcelaphus buselaphus*) and more (see Rey et al., 2017). Furthermore, Kuzikus is part of a national breeding project to enlarge the dwindling population of the Black Rhino (*Diceros bicornis*).

The data were acquired between May 12 and May 15, 2014, by the SAVMAP Consortium[2]. Five flight campaigns were conducted, and a lightweight single-wing UAV (SenseFly[3] eBee), equipped with a Canon PowerShot S110 RGB camera, was employed. A multispectral and a thermal sensor were also used to acquire data, but their lower resolution, coupled with the low temperature contrast during the day, made the annotation of the animals on such data unfeasible. During these campaigns the camera acquired a total of 654 images, each of size 4000×3000 pixels and 24 bit radiometric resolution. In total, this yielded more than 8.3 billion pixels, resp. an area of 13.38km^2, with an estimated average resolution of 4cm.

An initial ground truth (convex hull polygons of large animals) was provided by MicroMappers[4] (Offi et al., 2016) and consisted of 976 annotations in a subset of the 654 images, which was completed within three days. Since they were based on crowd-sourced volunteers efforts, some of the annotations were coarse or erroneous in that they occasionally omitted an animal, were not very accurate position-wise, or included multiple individuals in one annotation at once (Figure 2.2a). Even if the main target of this study was census-oriented, such blunders have detrimental effects on both the model's detection accuracy and the evaluation trustworthiness. For instance, sampling locations from the center point of a ground truth might result in the annotation lying between the animal and its shadow (*i.e.*, on the ground). All annotations were thus refined by the authors to only include the animal itself, and only one animal per location (Figure 2.2b).

We note at this point that due to the overlap between UAV images, it might occasionally be possible for one animal to appear in multiple images. Resolving this potential conflict would require matching heuristics that are beyond the scope of this chapter. For all

[1] `http://kuzikus-namibia.de/xe_index.html`
[2] `http://lasig.epfl.ch/savmap`
[3] `https://www.sensefly.com`
[4] `https://micromappers.wordpress.com`

(a) **(b)**

Figure 2.2: An example of annotations retrieved by MicroMappers from the crowd-sourcing campaign (left). Since the center points of each animal were used to train and test the models, all annotations had to be revised to only encompass individual animals themselves (right).

Table 2.1: Statistics of the Kuzikus 2014 acquisition. The data were split in an image-wise manner into roughly 70% for training, 10% for validation and 20% for testing purposes, based on the number of animals. To test for potential data biases, three different training/validation splits were carried out; for the training and validation sets average numbers of images and animals, together with their standard deviation (in brackets), are provided. The test set was identical in all three runs.

Set	#pixels	#images with/without animals	#animals
training	$5.83 * 10^9$ ($7.3 * 10^7$)	165.67 (6.5) / 291.00 (0)	834.00 (5.3)
validation	$8.90 * 10^8$ ($7.3 * 10^7$)	28.33 (6.5) / 41.00 (0)	114.00 (5.3)
test	$1.65 * 10^9$	45/83	235
total	$8.36 * 10^9$	415/239	1183

training and evaluations, only the center locations of the annotations were retained. The data were then split into training (70%), validation (10%) and test (20%) sets on an image-wise basis to ensure no annotation and no image were included in more than one set. Split priority was laid on the number of animals; all remaining empty images were likewise distributed according to the same percentages. Three different training and validation splits were created according to these rules, and the main models were trained on all three to avoid systematic biases due to the choice of splits. Due to the split rules and the uneven occurrence of animals in the images, the split statistics (number of images, number of animals, *etc.*) show slight fluctuations, but they are nevertheless within a reasonable range, of only a few percentage points, to each other. The test set is identical for the three cross-validation splits. The final dataset statistics are summarized in Table 2.1 and Figure 2.3. The images are publicly available at `https://zenodo.org/record/16445`.

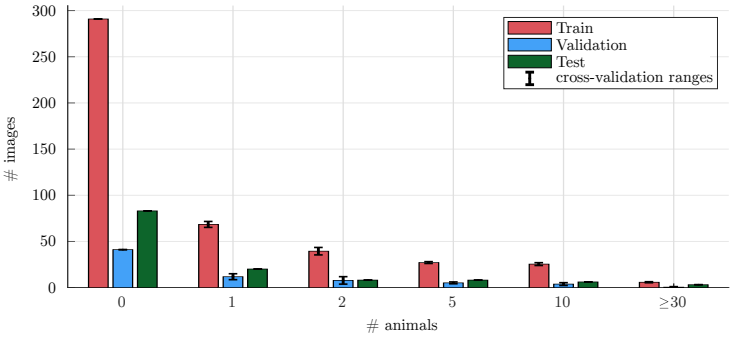


Figure 2.3: Distribution of the number of animals per image for the training, validation and test sets. Standard deviation ranges are given for the three cross-validation splits. The majority of images does not contain animals, which poses a significant challenge to any detector.

2.2.2 The challenges of covering large areas

Game reserves and national parks are typically concentrated areas, sometimes even confined, that contain wildlife in numbers above average, compared to the surroundings. In the case of the Kuzikus reserve addressed in this work, the estimated 3000 large animals live in an only moderately large reserve area. However, even under these conditions animals are a very rare sight, which makes livestock censuses a particularly arduous task. This is also the case of the dataset considered in this work, and locating an animal in UAV images corresponds to finding the needle in a haystack (Figure 2.4). As a consequence, UAV datasets can rapidly grow in size to dimensions that considerably hamper the performance of detectors.

The reason for performance degradation in large area sizes can be boiled down to two main problems:

- *(i.)* Increased background heterogeneity. In heterogeneous landscapes such as the African savanna, an increase in study area size will inevitably include substantially more landscape types, and hence more background variation. Human interpreters are aware of landscape variations and these usually do not cause any issue, but a detection algorithm that has been trained on one set of grounds is likely to fail on areas unseen.
- *(ii.)* Amplified coupling between the few animals and the background. This refers to the appearance heterogeneity within and homogeneity between classes. Animals themselves have a very diverse set of appearances due to different species, sizes, fur variations, and more. Detectors thus need to be able to learn all these variations as well as possible. At the same time, the expansion to more areas also means that

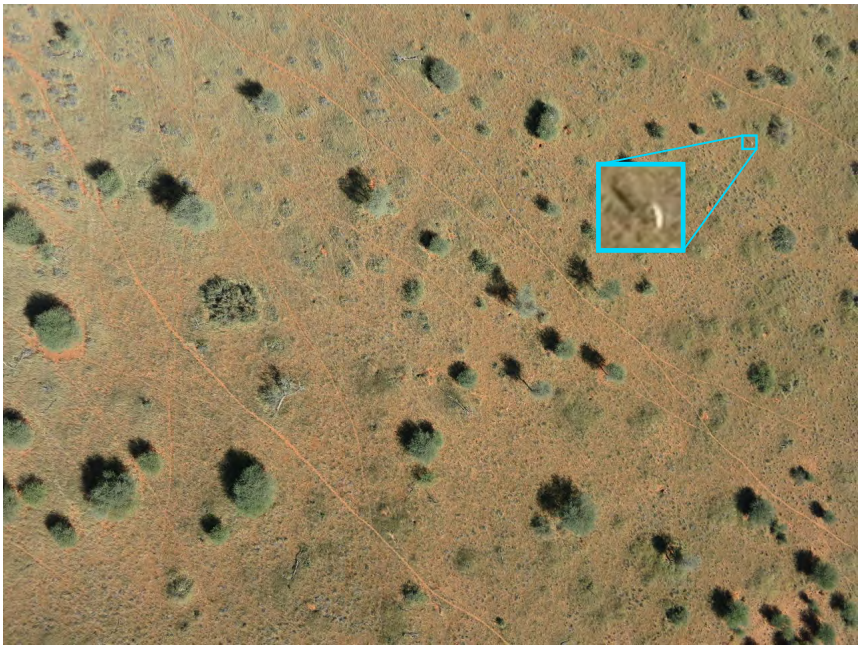


Figure 2.4: Full extents of a sample image. The single animal present in this scene (enlarged in the cyan bounding box) is difficult to locate, both for humans and machines.

a higher number of background objects looking like animals are included that may confuse a detector. Examples for such similar background objects are tree trunks and dirt mounds, as shown in Figure 2.1. A model that is trained to distinguish between individual animals is thus likely to mistake these background objects, and provide false alarms.

All these effects have the consequence that extrapolating results from a small to a big area is not going to yield trustworthy results, and models trained on a small subset and then evaluated on larger areas will not perform satisfactorily.

2.3 Addressing realistic and imbalanced datasets

In this section we discuss steps required to perform and improve semi-automated animal censuses on a realistic dataset. We begin by reviewing the working principle of CNNs in Section 2.3.1. In Section 2.3.2, we introduce a series of practices applicable during the training process of a CNN to make it learn all background variations while simultaneously learning the appearance of the small number of animals. Section 2.3.3 then defines the evaluation protocol we used.

2.3.1 Working principle of Convolutional Neural Networks

In this section we briefly present the concepts behind CNNs, but without ambitions to be complete. For an in-depth explanation, we refer to the comprehensive work of Goodfellow et al. (2016). CNNs are the workhorse of deep neural network methods in computer vision. Like other deep learning methods, CNNs allow to learn a set of hierarchical features that have been optimized for a given task (*e.g.* classification, regression, *etc.*). Their particularity consists in profiting from the inherent structure of images by extracting the features locally. Instead of using the whole image to compute a single feature, the same local feature is extracted in each image location (i.e each pixel), based only on the values from a few neighboring pixels. In particular, a convolution operation is used to extract the learned features, which generally considers a square neighborhood and captures a pattern of interest (*e.g.* a vertical green to brown transition) that is learned automatically. This means that, at each image location, we compute the scalar product between the square local neighborhood in the image and the pattern of interest, also called the *"filter"*. This set of local multiplications is called a *"convolution."* Convolving the filter at all locations in the image produces a so-called "activation map" that contains high values in the regions of the image where the pattern is more present.

After having extracted multiple such activation maps, this stack of maps (also called "tensor") can be treated as a new input image, from which new features can be extracted (*e.g.* a combination of green to brown transitions that indicate the presence of a tree in the image).

Since a composition of linear operators is also a linear operator, there is not much value added to composing a series of convolutions, other than it being equivalent to a single convolution with a larger filter (*e.g.* composing two convolutions with 3×3 filters can be equivalent to a single 5×5 convolution). To allow for richer, non-linear relationships between the input and the output, a simple non-linear function (*e.g.* a rectifier, sigmoid, *etc.*) is interposed between each pair of convolution operators. In addition to this, it is common to perform downsampling after some of the convolutions, which reduces the size of the tensor of activations and, therefore, increases the receptive field, *i.e.* the effective

area in the image that passes information to each local feature in the following layer.

CNN filters are generally initialized with random values or with values learned on some previous problem. In both cases, the filters can be improved for some particular task by defining a differentiable cost function that measures the goodness of the current solution. In our case, we employ the so-called "cross-entropy loss" that is standard for classification tasks:

$$\mathcal{L}(y, \hat{y}) = -\sum_{c} w_c y_c \log(\hat{y}_c) \tag{2.1}$$

In this loss, the prediction $\hat{y}$ of the CNN is compared to the true label y for every class c. Also note the weight w_c that amplifies or dampens the effect per class in comparison to the others, achieved by altering the learning rate itself depending on the true class for the respective sample.

For a given learnable parameter θ (e.g. an element in one of the convolutional filters), the gradient of the loss with respect to θ is computed. While this can be done directly for the last layer, the chain rule has to be applied for all hidden layers before, as their gradients depend on the gradients of the following layers. This update procedure is known as backpropagation and requires all operations used in the CNN to be differentiable. After computing the gradients with respect to each parameter, the parameters are updated with the following rule:

$$\theta \leftarrow \theta - \eta \frac{\partial \mathcal{L}(y, \hat{y})}{\partial \theta}, \tag{2.2}$$

where η is the learning rate.

Evaluating such function allows to improve the filters by gradient descent (Goodfellow et al., 2016). By applying this operation for multiple iterations, often several thousands, we can obtain convolutional filters for each of the layers that have been optimized for solving our problem. In sum, CNNs thus are not only able to learn a classifier, but also a series of expressive filters, produced by the convolutional operators, which allows them to yield superior performance in image-based classification tasks.

2.3.2 Training deep object detectors on imbalanced datasets

CNNs are commonly trained on large datasets containing thousands of examples for each class, and all classes often occur in similar percentages. For instance, the ImageNet dataset (Russakovsky et al., 2015) contains thousands of categories and more than 14 million images in total[5]. In our case, the overall amount of data is substantially smaller, but poses the additional challenge of data imbalance, with the background class being

[5]as of December 2019; `http://www.image-net.org`

overwhelmingly larger than the positive class, both in terms of quantity and sample complexity.

As a result, a CNN trained without specific alterations in the training plan on such a naturally unbalanced dataset will be flooded by background examples, and will inevitably miss most animals.

In order to account for these problems, a CNN must be trained in such a way that it learns all possible variations in the background, while also addressing the appearance of the few foreground instances it has seen. The entire process corresponds to finding an equilibrium between two extremes (not detecting the animals at all or firing false alarms everywhere), which can be very delicate. In the following, we present a series of recommendations applicable during CNN training that will steer the model in the right direction. The effects of each individual procedure will be assessed in an ablation study, whose results are presented in Section 2.4.2.

Addressing the class imbalance

An initial solution to overcome the imbalance problem might be to artificially balance the dataset by oversampling, *i.e.* repeating each animal instance to match the total number of background locations. While this has been shown to work well for other tasks (Buda et al., 2018), we empirically found it to cause the CNN to overfit to the small number of animal instances present in the training set. The inverse, *i.e.*, reducing the number of background samples to match the number of animals ("undersampling") has a similar effect in that the model fails to learn the variability of the background, which leads the model to misdetect everything that looks even remotely similar to animals.

We instead apply conventional class weights to reduce the impact of the background class. In practice, we set the class weights w_c in Eq. (2.1) to different values, to ensure that an error in the animal class counts much more than one in the background class. Exact values depend on the data value range, the training schedule and the amount of data the CNN is trained on and may thus vary from problem to problem. Nevertheless we got satisfactory result by weighting classes according to the inverse of the frequency in which they occurred in the training set.

Making the model learn all background variations

Large natural datasets contain a lot of variations in appearance for both the animals and background classes. This causes an increased number of false alarms. To avoid such confusion, the CNN needs to be trained on as much of the background area as possible. Below we present techniques that can employed to achieve such goal.

We start with the observation that, thanks to their convolutional nature, CNNs can process inputs of arbitrary spatial dimensions, up to the limits of the available memory. This has the effect that CNNs can be trained not only on single patches around specific locations, but on larger images. In such a setting, evaluating on a larger image will not yield a single label, but a spatial grid of activations. An advantage of proceeding this way is that neighboring locations partially share activations of overlapping filters, and this way a much higher data throughput per training iteration can be achieved, as less redundant filter activations need to be stored for the backward pass. Using larger training images therefore has a similar effect to increasing the number of patches, while being more memory-efficient. Moreover, it allows the model to make full use of its receptive field, thus capturing as much context as possible and reducing the overall impact of border effects. Nevertheless, this size is limited by amount of memory available on the GPU for storing the intermediate activations. We achieved satisfactory results by training models with image patches of size 512×512 pixels, cropped from the larger images at semi-random locations: if a training image contains animals in the ground truth, we crop the patch so that it encompasses at least one animal instance. In images without an animal, the location is randomly chosen at every iteration to maximize the learned background diversity.

Curriculum learning as a starting boost

CNN training corresponds to a series of epochs in which portions of the training dataset are fed to the network in random order. Consequently, it is an iterative process. Since the training samples picked during each iteration might only reflect parts of the full dataset distribution, the ordering can be very decisive in the quality of the final model. Although the model parameters are constantly being adjusted, chances are that it "forgets" the representation of certain instances once the learning process shifts to other parts of the dataset. In our case, this risk is related to the class imbalance: if the CNN is trained on the full dataset from the start, the background class might drown the signal emerging from the animals, despite the class weights (see Section 2.3.2). We propose to countersteer this effect based on the concept of curriculum learning (Bengio et al., 2009), where a model is adaptively trained with different portions of the dataset. While curriculum learning is commonly applied in cases where datasets show gradually increasing complexity, we employ it to force the CNN to learn a more balanced representation of both animals and background in the beginning. To this end, we start training the model on a subset of the training data that has been artificially restricted to image patches that contain at least one animal. This way, the CNN has the chance of learning how animals should be represented, while having only an approximate knowledge of the variability of the background. We use this schedule for five epochs, and then switch over to the full dataset, where image patches might not always have an animal. After this switch, the model is confronted to the entire variability of the background class.

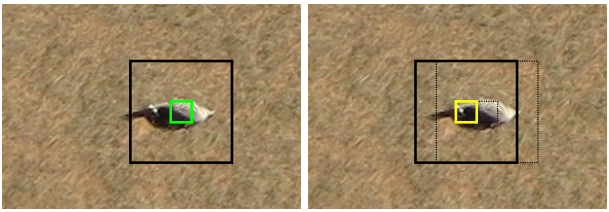
Rotational augmentation

One practice commonly employed during the training of CNNs is *data augmentation*. This consists of creating artificial training samples by altering the existing one in realistic ways. A common augmentation strategy is to rotate images randomly during training. Since our UAV dataset has a perpendicular top-down perspective, rotating the image has a similar effect to acquiring the ground with the UAV at a different angle. In theory, applying rotations at random to images should make the detector more robust to grounds and animals occurring at different orientations in the field. In other words, rotational augmentation allows us to explore a larger set of possible viewing angle versus animals configurations.

However, a critical side-effect of applying extensive augmentation is that it significantly changes the dataset seen by the model itself. For example, if the training images are rotated at random with a probability of 50%, the model sees up to 1.5 times as many data points as before. In the case of rotations, where usually every pixel is altered significantly, this might lead to too strong perturbations. Especially at the beginning of training, a model is supposed to get an overall glimpse of the statistical distribution of animals and background in the dataset. Instead, we limit data augmentation to random horizontal and vertical flips of the images with a 50% probability each during the first 300 epochs. Only afterwards, we introduce random rotations with a 75% probability and train the model for another 100 epochs with a at this stage slightly reduced learning rate. We limit rotation angles to multiples of 90 degrees to avoid spatial shifts of the (reduced size) ground truth due to nearest-neighbor interpolation. More advanced augmentation techniques, such as synthetic data generation using conditioned generative adversarial networks (Shrivastava et al., 2017; Radford et al., 2016) are exciting perspectives of our system that might be considered in the future.

Hard negative mining

The general idea of class weights and curriculum learning explained above is to prevent one class from dominating over the other. This is especially important at the beginning of the training procedure, where the model parameters are not yet optimized enough to the problem at hand. However, after a sufficient number of iterations, a state will usually be reached where the model performs fairly well in finding the animals, but might not reach its maximum precision. At this stage, the model performance can be further boosted to reduce the number of false alarms by shifting from treating all data points equally to focusing on the errors. We propose to do so by means of hard negative mining (Malisiewicz et al., 2011; Shrivastava et al., 2016). This technique essentially gives special treatment to the most severe mistakes, which in our case are the false positives scored with the highest confidence. In practice, we train the CNN regularly for 80 epochs and then employ hard



(a) Receptive field on an animal **(b)** Receptive field at the border

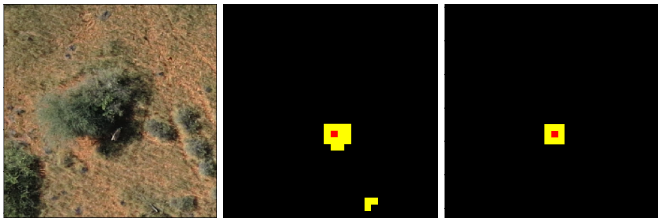
Figure 2.5: A CNN takes into account multiple neighbors of a specific location ("receptive field") to determine the likelihood of the center location being an animal (left). To avoid confusion of animals not in the center, but still in the receptive field (right), we introduce a border class around a true location.

negative mining by locating the four background samples in the 512×512 grid that have been scored with the highest confidence, and assigning them a dedicated weight that is ¼ of the animal class weight. We do not assign more weight to prevent the model from forgetting the animal appearances, as our ultimate goal is to maintain a high recall (see Section 2.3.3).

Introduction of the border class

A slightly less obvious problem of training a model on larger patches instead of single examples is the effect of *"spillage"* along the spatial boundaries between locations of different classes. CNNs classify locations (in our case, into "animal" *vs.* "background") by taking into account the location's neighboring pixels via the model's receptive field. For instance, in Figure 2.5a, the location of interest (in green) is predicted based on the CNN's receptive field size (in black). While this works as intended at the center of an animal, it might fail in its surroundings (Figure 2.5b): in this case, the final label should be "background" (yellow square) to avoid multiple predictions of the same object. However, the CNN's receptive field still includes a major portion of the animal itself. Training using the latter case risks to produce the undesirable effect of the CNN learning that patches influenced by animals through the receptive field belong to the background class. Ignoring these locations in turn (setting the weights to zero) will give the model too much freedom in exact positioning, which might lead to false alarms in the surroundings of the center location. This is particularly problematic with animals standing close-by, as they cannot be properly separated anymore.

An initial solution to this problem is to decrease the CNN's receptive field in order not to include nearby animals. This however has the undesirable effect of providing a too fine grid of label predictions. As a consequence, an animal may suddenly encompass multiple



(a) Image patch **(b)** Model prediction **(c)** Ground truth

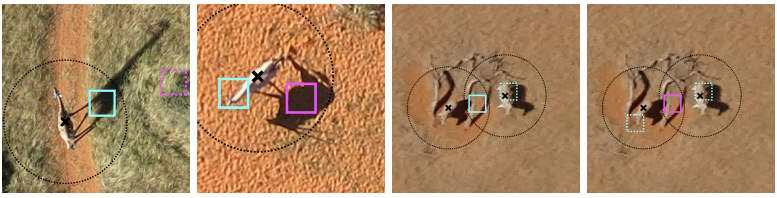
Figure 2.6: Example image patch (left), its prediction grid by the CNN (middle), and the corresponding ground truth (right). As can be seen, the CNN learns to distinguish between background (black), animals (red) and the border class (yellow) fairly well.

locations, causing multiple neighboring false alarms that need to be suppressed (Kellenberger et al., 2017a). Furthermore, a too small receptive field will make the CNN fail in capturing the full appearance of the animal.

We propose to address this issue by instead including a dedicated "border class," which corresponds to locations that technically belong to the background, but still include portions of an animal. The effect of a border class is that the CNN learns to treat spatial transition areas separately and predict them accordingly. At test time, locations predicted as "border" by the CNN can simply be discarded, leaving the predicted center location as the only detection. Multiple ways of including a border class are possible and depend on the sizes of the animals and CNN prediction grids. In our case, we obtained satisfying results by assigning the eight neighboring pixels around an animal center to the border class (see Figure 2.6).

2.3.3 Census-oriented evaluation protocol

Although many census-based studies include evaluations using common metrics on a held-out test set, they do not explain the exact criteria required by a detection to be counted as a correct match. While this is well established for classification (a data point is hard-assigned to a class), it is less clear for object detection. For instance, if the main target is not to retrieve the exact position of animals, the question of how far away predicted animals can be from the ground truth locations arises. Traditional computer vision measures, such as the Intersection-over-Union (Everingham et al., 2010), are intended for spatially precise object detections and would penalize slight spatial deviations too harshly for our application. Also, the handling of multiple predictions for the same animal is generally not explicited. This is especially important in the case of multiple animals standing close-by each other.



(a) (b) (c) (d)

Figure 2.7: Example cases of the census-oriented evaluation protocol. Predictions outside a certain distance range (2.7a) as well as multiple predictions of the same animal (2.7b) are rejected (magenta). In the case of a prediction matching more than one animal, it is counted as a true positive (cyan) if at least one animal is not covered by another prediction (2.7c), or marked as a false positive if all are (2.7d).

To account for these problems, we propose an evaluation protocol that reflects the final objective: to provide animal abundances, whereas the exact pixel location is only secondary. This allows us to judge the quality of each prediction according to the following set of rules:

- Predictions may be candidates for correct detections if their position falls within a (circular) range of reasonable size around each ground truth's center location.
- A ground truth object is counted as correctly identified if at least one prediction falls within the distance range.
- n predictions inside the distance range are counted as one true positive, and $(n-1)$ false alarms.
- If a prediction lies within the distance range of more than one ground truth location, it may be counted as at most one true positive, given that there is at least one ground truth object that has no other prediction in its range. If all involved ground truth objects are already predicted by other predictions, the current one is discarded as a false positive.

Consider the examples in Figure 2.7. In situation 2.7a, there are two detections in vicinity of a single giraffe, but both of them miss it spatially by quite a margin. However, the one closer to the animal (cyan) actually detects its shadow, and should thus be counted as a true positive. The other (magenta dashed) lies outside the detection range and is thus considered a false positive. Situation 2.7b shows one animal surrounded by multiple predictions, all of which lie inside the allowed distance range. Since our main target is to provide animal counts, only one prediction (generally the closest) should be counted as a correct detection (cyan), and the others dismissed as false alarms (magenta). However, such a rule cannot always be applied straight away, as situation 2.7c shows: here,

two animals are surrounded by multiple detections, out of which one falls directly in between the two (cyan). In this case, the prediction in-between could be counted as a false alarm considering that the animal to the right is already predicted by another detection (cyan dashed). However, as the other animal has no other detection in its vicinity, both predictions should be treated as correct in this case. If, however, all involved animals are already covered by an equivalent number of predictions (case 2.7d), any superfluous prediction, such as the one in the middle (magenta), is marked as a false positive. In all cases, animals completely missed by the detector are counted as false negatives.

The effect of such a procedure is that the number of true positives cannot exceed the actual number of animals, which corresponds to a realistic census scenario. A hypothetical ideal detector will reach a perfect score if the number of predictions matches the number of actual animals, and if they are reasonably close to the ground truth positions. The only parameter to be set manually is the distance range around ground truth objects, and we will evaluate effects of different ranges in Section 2.4.4. The protocol is nevertheless robust enough even at less reasonable ranges: if the distance range is set too small, the penalization is increased and only detectors that provide a good enough positional accuracy will yield good scores. In turn, an unreasonably large distance range will include more potential predictions further apart, but multiple detections are still dismissed as false alarms. In all cases, the protocol will assign the best scores if the number of predictions matches the number of animals present in the scene, which is the ultimate goal of census campaigns.

2.4 Experiments and results

In this section we present the effects of the training practices described above. We start by describing the models we used (baseline and CNN) in Section 2.4.1. The CNN training recommendations, including rotational augmentation, are then put to test in a dedicated number of ablation studies (Section 2.4.2). A third chapter addresses the effects of choosing different evaluation distance thresholds (Section 2.4.4). Finally, Section 2.4.5 presents the results obtained on the full dataset based on the best baseline and CNN architectures, respectively.

2.4.1 Models setup

CNN with proposed training schedule

We base all experiments on a single CNN architecture, a simplified version of (Kellenberger et al., 2017a) that performs detection, but no regression of bounding boxes around the animals. The overall architecture is shown in Figure 2.8. The base feature extractor consists

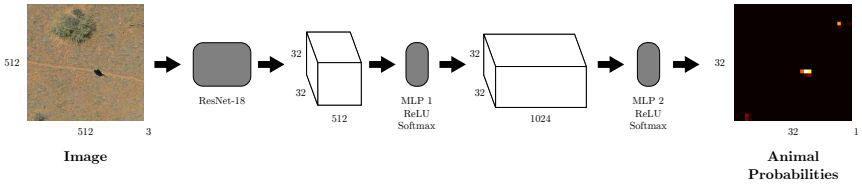


Figure 2.8: Working principle of the CNN-based animal detector. We train the model on images of size 512×512, which results in a grid of 32×32 locations with animal probability scores for each location.

of the first convolutional and four subsequent residual blocks of an instance of ResNet-18 (He et al., 2016). This model had been pre-trained on the ImageNet classification challenge (Russakovsky et al., 2015), *i.e.* has been trained on the task of classifying visual classes that are not specific to wildlife monitoring. To enforce the model to be specific to our problem, we adapt it by adding two one-by-one convolutional blocks or Multi-Layer Perceptrons (MLPs) with nonlinear activations (Rectified Linear Unit (ReLU)), dropout regularization (Srivastava et al., 2014), and softmax activation at the end. This way of proceeding, also called *fine-tuning*, is a very common strategy used to adapt these models to specific applications, since generic deep learning architectures (such as ResNet) are very good at extracting relevant generic local features in the earlier layers and only need to be made specific to the problem at hand in the deeper layers (Castelluccio et al., 2015).

Table 2.2: The CNN models trained with different recommendations: for the ablation study, five models were trained with one of the recommendations held out (top rows); the final model (bottom row) used all recommendations. "Full Dataset" denotes models were trained on 512×512 patches both with and without animals; CNN 2 only saw patches that contained at least one animal. The final model ("Full Model") corresponds to CNN 6, further fine-tuned with rotational augmentation from epoch 301 to 400.

Model	Class Weights	Full Dataset	Curr. Learning	Hard Negatives	Border Class	Rot. Augm.
CNN 1		✓	✓	✓	✓	
CNN 2	✓		✓	✓	✓	
CNN 3	✓	✓		✓	✓	
CNN 4	✓	✓	✓		✓	
CNN 5	✓	✓	✓	✓		
CNN 6	✓	✓	✓	✓	✓	
Full Model	✓	✓	✓	✓	✓	✓

Table 2.3: Parameters for the recommendations (where applicable), used to train the full CNN.

Recommendation	Parameters
Class weights	1.0 (animals), 1/80 (background), 1/8 (border)
Curriculum learning	5 epochs only on images with animals
Hard negative mining	weight 0.5 for 4 hard negatives (from epoch 80)
Border class	over 8-neighborhood around animal
Rot. augmentation	75% random 90° stops [±270°] (from epoch 301)

To assess the training recommendations presented in Section 2.3.2, we trained a series of CNNs in a held-out fashion: for each model, we enable all but one of the five main recommendations presented above (and summarized in Table 2.2). The specific properties and parameters of the recommendations are given in Table 2.3. The final model (referred to as "Full Model") had all recommendations enabled. We separately analyzed the effect of rotational augmentation in a dedicated ablation study (Section 2.4.3). All models were trained using the Adam optimizer (Kingma and Ba, 2014), with momentum of 0.9 and a learning rate gradually decreasing from 10^{-4} (first 5 epochs) to 10^{-5} (next 5 epochs), to 10^{-6} (next 100 epochs), and then again to 10^{-7} for the rest (until final epoch 400). Weight decay helped reducing overfitting effects and was employed with a factor of 10^{-3} for the first five epochs, and 10^{-4} for the rest.

At test time, we slid the model over each whole image and retrieved a prediction grid coarser in resolution, with confidence scores for every grid cell (and hence location) in the image to contain an animal. In detail, we divided each 4000×3000 image into 8×6 subpatches of size 512×512 and evaluated each patch individually, which yielded a prediction grid of 32×32. Note that the patches did overlap at their borders to some extent; in those areas we eventually averaged the obtained class probabilities when stitching the patches together. Eventually, this yielded a 188×250 prediction grid for an entire UAV image.

All CNN-based models were implemented in PyTorch[6]. Training time per model took approximatively four days on a Linux workstation with an Intel Xeon CPU and an NVIDIA GeForce GTX 1080Ti graphics card.

Baseline model

As a baseline, we employ the current state-of-the-art on the Kuzikus dataset presented in Rey et al. (2017). This model relies on a pre-selection of candidate locations ("object proposals"), which are then classified in a second step. We use the same object proposals

[6] `http://pytorch.org`

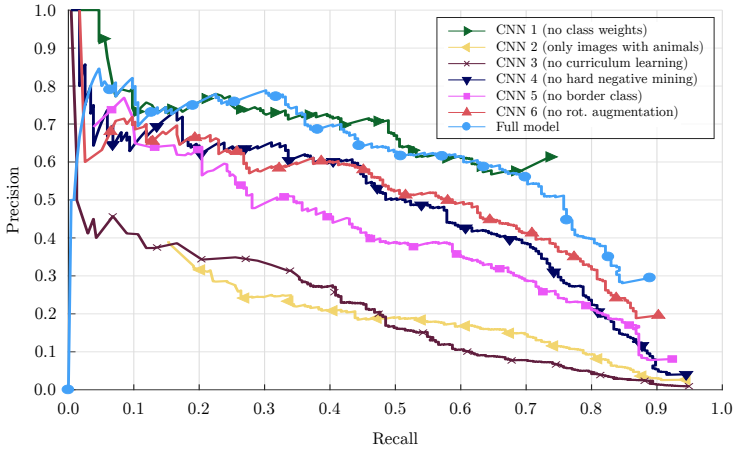


Figure 2.9: Precision-recall curves for the different CNNs on the test set, based on the CNN confidence scores. All recommended training recommendations provide an increase in precision at given recall values, but the best model emerges when all of them are combined together, and when the model is further fine-tuned using 90 degree-stop rotational data augmentation.

and feature extraction methods proposed in the original work, but replace the original exemplar SVM with a random forest classifier, which is a solid baseline in remote sensing, providing nonlinear response at no extra computational expense (Pelletier et al., 2016). We found the random forest to provide a similar performance as reported in Rey et al. (2017), which can also be seen in the precision-recall curves in Figure 2.12.

2.4.2 Ablation studies

In this section, we first evaluate the impact of the five basic recommendations for effective CNN training provided in Section 2.3.2, followed by a study on rotational augmentation. Finally, we also assess the role played by the distance threshold in our assessment procedure.

Effects of the recommended CNN training recommendations

Figure 2.9 illustrates the results obtained by the CNNs described in Section 2.4.1 on the complete test set. As can be seen, almost every aspect of the proposed CNN training plan in itself provides an improvement in precision, but mostly in regions of high recall (beyond 80%). The best model is the one trained with all recommendations applied together. The

effect of each individual recommendation does not seem to scale linearly, but rather to depend on the enabling of the other recommendations.

Quite surprisingly, "CNN 3," the model trained without curriculum learning (that is, a model trained with the full training set from the beginning) yields the worst precision at high recall values. The CNN 2 model, trained only with images that contain animals, also performs poorly and only marginally better than CNN 3, therefore implying that only including more background data from the start to train the CNN is not sufficient. Instead, the dominating background class forms a hurdle that can only be overcome by injecting more background patches, intertwined with curriculum learning. This could be explained with the training behavior of CNNs at early epochs: at the beginning of the training, the CNN weights are not yet set to values where their prediction yields a (local) optimum, that is, a low value in the loss function. The model requires several passes over the dataset for the weights to become meaningful for the problem at hand. In an imbalanced setting as mammals detections in the wild, this initial phase appears to be especially critical, as the model needs to learn sufficient data about *both* the background and animals classes. Once this trade-off between the two classes is reached, the models can then be fine-tuned to learn more background variations, thus harnessing all the potential of curriculum learning.

Confusion can further be dampened if the border class is introduced: in fact, the model trained without the border class ("CNN 5" in Figure 2.9) yields the third-lowest performance on the test set and performs especially poorly at high recall rates, where approximately 19 false alarms are recorded per positive example (5% precision). We argued above that the border class helps in reducing confusion as the CNN still includes parts of the animals nearby due to its spatial receptive field. Although slightly more CNN parameters need to be learned with every additional class, the results indicate that the border class does indeed guide the CNN to a better solution by distinguishing between animal centers and locations surrounding them.

The model that omitted hard negative mining ("CNN 4" in Figure 2.9) performs similarly to the full model. Such behavior could be expected, given that hard negative mining mostly attempts at boosting the precision at very late training stages. We observe that the technique comes into play at the most critical regions of 90% recall and more, where it manages to increase the precision from around 5% (CNN 4) to 20% (full model). In numbers, this results in a reduction of false alarms from 19 to 4 for every detected animal. While curriculum learning helps in initiating the model in the best way, hard negative mining takes over at the final fine-tuning stage. Clearly, both techniques seem to play a major role in improving the model performance, and thus reducing the amount of time required to manually verify the detections, as we will discuss in Section 2.4.5.

Perhaps the most surprising result emerged from the model that was not trained with balancing class weights ("CNN 1" in Figure 2.9). We repeated this schedule several times

and observed the same behavior in all runs: the model would predict every location as "background" for about 45 to 55 epochs, but would then suddenly raise to a precision of almost 100% and a recall of around 40% in the training and validation sets—all within one epoch. The model would then improve on the recall over the remaining epochs. Once the model learns to correctly predict the background class (which is the majority of the samples), the only error signals would emerge from the few animal instances, making the model focus on them. However, the definite cause for the model to jump from one extreme state to another in just one epoch still needs to be resolved. In any case, we trained this model for the full 300 epochs like the others, and although it reaches superior precision for the most part, it fails to exceed in recall over around 74% of the animals. In other words, this model would never be able to retrieve the remaining 26% of the animals, no matter the detection threshold selected. In livestock estimations, missing over a quarter of the animals might be fatal. We hypothesize that this corresponds to a substantial overfitting effect to the training set, which means that this model is by far not apt to be used in real census scenarios despite being the one providing the highest precision for lower recall rates.

2.4.3 Rotational augmentation

As explained in Section 2.3.2, we employ random rotations as one of the data augmentation strategies. We decided to assess rotational augmentation in a separate study, as we found it to have remarkably fluctuating influences on the final model performance, depending on the training stage at which rotational augmentation is employed. In detail, we experimented with three different training schedules:

- Training a model with all recommendations and rotational augmentation from the beginning;
- training with all recommendations, but enabling rotational augmentation from epoch 150;
- training with all recommendations, but enabling rotational augmentation from epoch 300 for another 100 epochs.

The resulting precision-recall curves for the these models can be seen in Figure 2.10. Compared to the initial model that was trained with all recommendations except for rotations (pink), two opposing effects of rotational augmentation can be observed, depending on the stages it was employed. If rotations are enabled at relatively early phases in the CNN training, such as epoch 150 (green) or even from the start (red), the final accuracy of the model is similar or worsens: for similar recall values, lower precision figures are observed. As it seems, rotations do have an influence significant enough to slightly confuse the detector at these early stages. If, however, the model is only fine-tuned from epoch 300, also with a lower learning rate, we see a substantial improvement in precision, with only

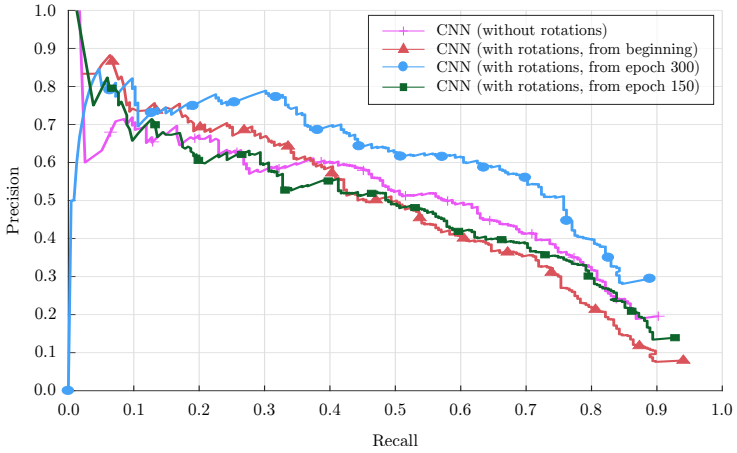


Figure 2.10: Precision-Recall curves on the test set for the ablation study on rotational augmentation.

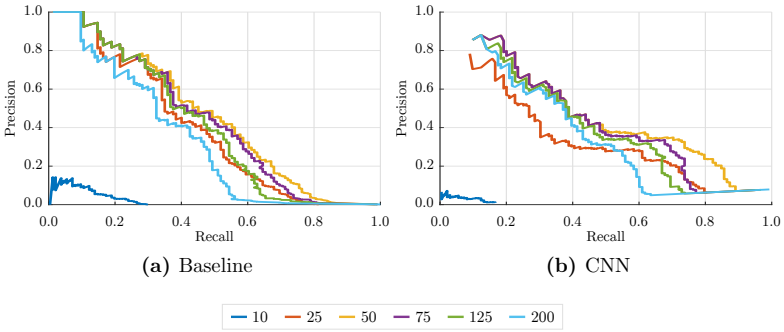
a slight sacrifice in overall achievable recall. Based on these findings, we may conclude that rotations are indeed a valuable augmentation strategy, when applied at later training stages and with gentler learning rates.

2.4.4 Evaluation of the distance thresholds

In Section 2.3.3, we introduced an evaluation protocol that works by addressing predictions within a certain distance range that has to be manually set. Sensible thresholds depend on the data resolution, target sizes, and positional accuracy of the model predictions. In this section, we assess the impact of different evaluation thresholds on predictions from both the baseline (random forest following the protocol of Rey et al. (2017)) and CNN models.

Figure 2.11 shows the precision-recall curves for both models on the validation set, evaluated at thresholds ranging from 10 pixels (0.4m) to 200 pixels (8m). Several common trends can be observed, such as the weak performance at the smallest threshold. This implies that both models struggle in positioning their detections precisely on the exact location of the ground truth point. While this can partially be attributed to suboptimal model performance, another potential cause is the limited precision in the ground truth itself, which may come from imperfect animal center delineation, detections on shadows, and the like.

At larger thresholds, the protocol is more tolerant towards spatial shifts. Consequently,



(a) Baseline **(b)** CNN

(c) Legend (distance thresholds in pixels)

Figure 2.11: Precision-recall curves for the baseline (left) and CNN (right) on the validation set, evaluated with different distance thresholds. Too small thresholds like 10 pixels (40cm; dark blue) require a high positional precision which neither model can deliver; too large thresholds like 200 pixels (8m; light blue) include too many false alarms.

the curves improve significantly. The baseline seems to score best at around 50 pixels, whereas the CNN has the best precision at 25 in the interesting region of high recall values. Furthermore, the absolute precision values at identical recalls are higher in case of the CNN compared to the baseline. Both these circumstances indicate that the CNN manages to provide predictions that are spatially closer to the actual ground truth.

Once the evaluation threshold exceeds a certain limit, the protocol includes predictions that are simply too far away from an animal to be accounted as a valid prediction. At the same time, multiple predictions of the same animal are discarded as false positives, which has negative effects on the precision. A look at the figures confirms that the precision-recall curves start to drop as soon as the distance threshold gets unrealistically high.

Following this ablation study, we decided to use a distance of 50 pixels (around 2m) in the results presented in the next section.

2.4.5 Results on the full dataset

Animal instance-based evaluation

We evaluated the full models on the test set and assessed their performances using the census-oriented evaluation protocol discussed above. All hyperparameters were chosen according to the best performance on the validation set (see Section 2.4.1 and Table 2.3 for details).

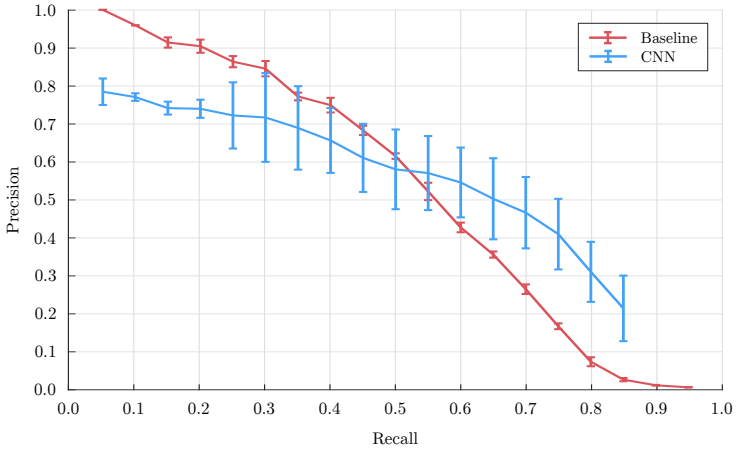


Figure 2.12: Precision-Recall curves on the test set for the baseline (red) and proposed CNN (blue), evaluated at a distance threshold of 50 pixels. Vertical bars denote the standard deviation for both models among the three cross-validation splits. The proposed model manages to significantly reduce the number of false alarms at high recall rates.

Figure 2.12 shows the obtained precision-recall curves for both models; Table 2.4 lists statistics for recall rates of 70 and 80%, respectively. Values are provided for ranges reachable by all three respective cross-validated models. We note that the model variations along the three cross-validation splits are very low for the proposed CNN, and particularly controlled for the baseline. This indicates that any biases emerging from the choice of images in the dataset splits are virtually negligible. Compared to the baseline, the CNN trained with the proposed recommendations ("Full model") yields a significantly better precision at high recall values, which can be traced back to a greatly reduced number of false positives. In other words, the CNN yields similarly high recall values while making less mistakes. Numerically, the reduction in number of false positives is striking: from over 2500 to less than 450 if the target is to find 80% of the animals (Table 2.4). For a recall level of 90%, our full model produces 870 detections, whereas the baseline scores a staggering 20'688. At such high recall values, it is unavoidable that models will struggle and provide false alarms, especially given the problems induced by covering large areas. However, looking at the results in a visual example (Figure 2.13), it becomes clear that the proposed CNN detects most animals limiting the mis-detections, while the baseline predicts animals almost everywhere, making the work of human annotators more time-consuming. On the contrary, the CNN detections can be rectified by a trained annotator in a limited amount of time. At a target of 90% recall, the baseline produces at least 51 and up to 773 detections per test image, even though 83 of the images do not contain animals

Table 2.4: Model performances at different levels of recall, averaged over the three cross-validation splits (TP = true positives; FP = false positives; FN = false negatives; F1 = harmonic mean of precision and recall). The proposed CNN manages to substantially reduce the number of false alarms, and with that the number of tiles to be screened ('Num. tiles'). Note that due to the confidence behavior of both models and the different splits the exact numbers of true positives slightly differ.

Recall	Model	TP	FP	FN	Precision	F1	Num. tiles
0.7	Baseline	164.3	459.7	70.7	0.3	0.4	281.0
	CNN	164.0	196.7	71.0	0.5	0.6	187.0
0.8	Baseline	188.3	2546.0	46.7	0.1	0.1	942.3
	CNN	188.0	447.3	47.0	0.3	0.4	268.0

at all. The CNN in turn predicts no animal correctly in 34 images and a maximum of only 81 detections in one image. In other words, this means that the CNN greatly reduces the effort required to verify the predictions.

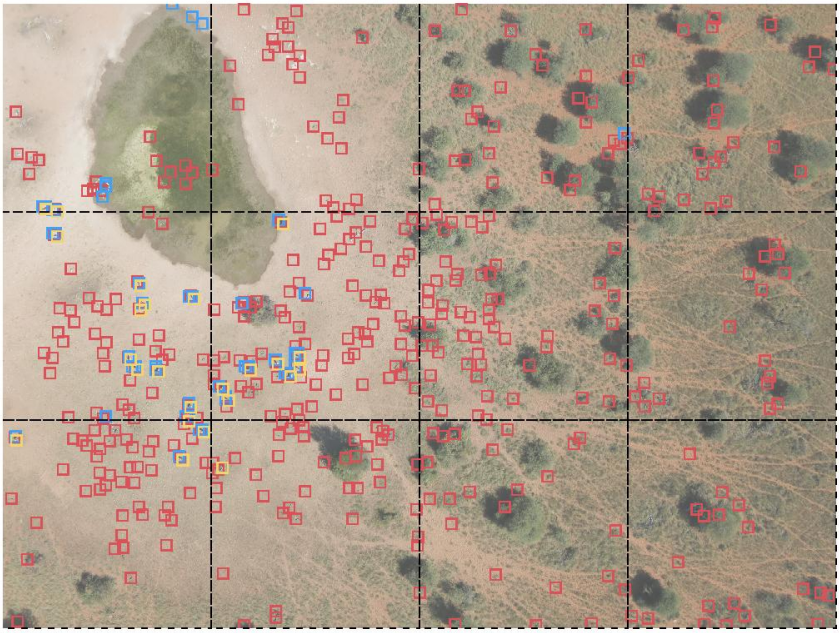


Figure 2.13: Prediction results on a test set image. Both models were set to yield 90% recall on the test set. The CNN (blue) manages to produce far less false alarms compared to the baseline (red). Ground truth locations are shown in yellow.

Tile-based evaluation

In manual verification settings, the raw UAV image size (4000×3000 pixels) prohibits a complete animal identification by naked eye, given the small target size of only a few pixels (Figure 2.4). It is thus common practice to split up the image into regular tiles and assess each one individually. If we assume a tile size of 1000×1000 pixels as a realistic size for naked eye screening, the human operator would need to screen 12 tiles per full image acquired. Based on this assumption, we can calculate further statistics tailored towards model aptness for human post-processing. A good model will only predict locations in tiles that actually do contain an animal; less precise models will also fire in tiles without animals. This also means that we have hypothetical boundaries for a perfect model (a model that only scores in the tiles containing animals) and a lower bound (a model that scores in all tiles). In our test set, these figures lie at 97 and 1536 tiles, respectively.

Figure 2.14 shows the number of tiles versus tile-based recall for both the baseline and

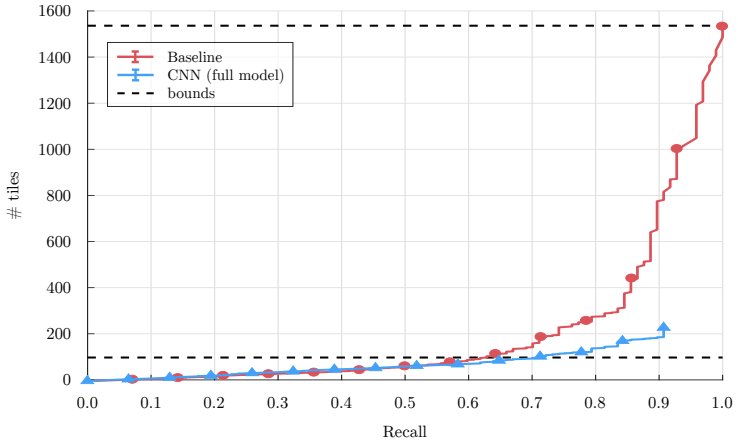


Figure 2.14: Number of 1000×1000 tiles with detections in comparison to a tile-based recall, given for both models on the test set. The total number of tiles in the test set was 1536 (upper dashed line); ground truth objects were found in 97 tiles (lower dashed line).

the CNN. Specific results at fixed recall values are provided in Table 2.5. On average, the proposed CNN detects animals in a lower number of tiles, while being able to concentrate predictions to the relevant ones (high tile-based recall). For instance, the CNN manages to find 90% of the tiles containing animals (tile-based recall of 0.9), while only producing a total number of 190 tiles that need to be verified (93 more than theoretically needed). The baseline detects spurious animals in 779 out of the hypothetical 1536 tiles instead.

Table 2.5: Tile-based results at different recall levels, averaged over the cross-validation splits. Tiles may encompass more than one ground truth animal; this results in overall higher precision scores compared to the individual animals evaluation presented in Section 2.4.5.

Recall	Model	TP	FP	FN	Precision	F1	Num. tiles
0.7	Baseline	68.0	95.0	29.0	0.4	0.5	163.0
	CNN	68.0	41.0	29.0	0.6	0.7	109.0
0.8	Baseline	77.7	200.0	19.3	0.3	0.4	277.7
	CNN	78.0	77.0	19.0	0.5	0.6	155.0

Finally, Figure 2.15 shows the number of tiles against recall, based on the individual animals. This case can be seen as a combination on the annotator's target (*i.e.*, finding all animals) and the way to get there (verifying a certain set of tiles). Although the curves share similarities with those in Figure 2.14, there are still notable differences: the CNN curve slightly plateaus in the number of tiles at high recall levels, which indicates that once a certain threshold is exceeded, additional detections do not spread out more in space.

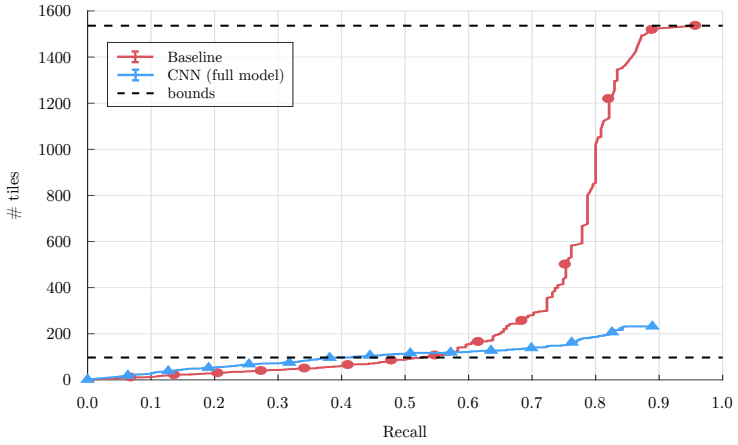


Figure 2.15: Number of 1000 × 1000 tiles with detection for a given recall, reported on the test set.

At such recall values, the baseline produces at least one prediction in almost every tile and therefore reaches the hypothetical maximum of 1536 tiles. Visualizing one of the test images (Figure 2.16) confirms this intuition. Summing up, the proposed CNN limits its detection to the relevant image tiles, thus easing the workload of human operators.

2.5 Conclusion

In this chapter we discussed semi-automated models that work on UAV images and assist human operators in counting large mammals over vast areas, such as the African savanna. This task is of central importance to animal conservation, as manual counting and photo-interpretation are prohibitively time-consuming. We focused in particular on the problem caused by the scarcity of animals with respect to the amount of images acquired in a campaign. Even a good model, one detecting with high precision on a selected subset of the data that is known to contain animals, tends to provide an enormous amount of false positives when applied to parts of the dataset containing swaths of empty savanna.

We proposed and discussed a methodology for animal censuses based on a CNN and showed how to train deep animal detectors on a real-world UAV image dataset consisting mostly of images with no animals. We introduced several recommendations to guide a CNN in the right direction during training. These include *(i.)* class-weighting to soften the impact of the overly abundant background class, *(ii.)* curriculum learning to prevent the model from forgetting animal representations, *(iii.)* hard negative mining to

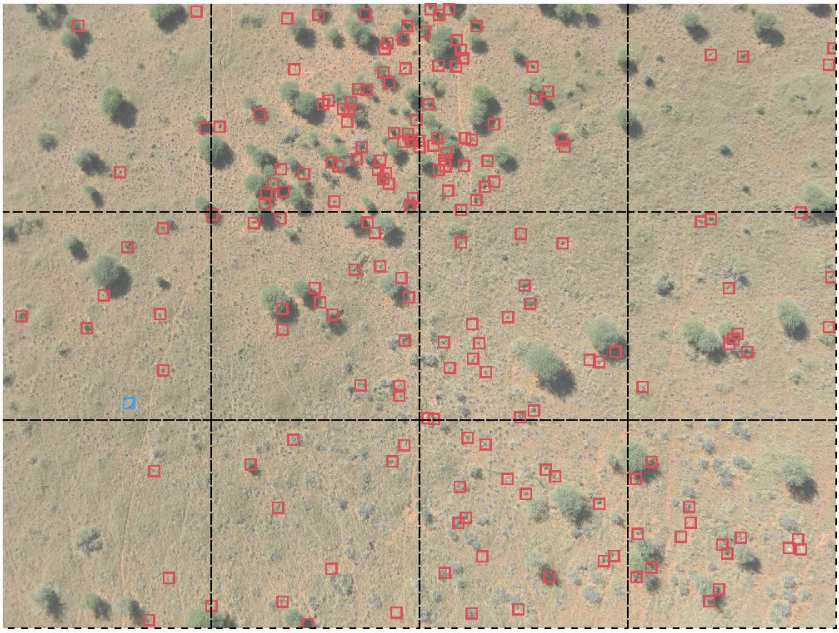


Figure 2.16: Predictions on a test image that does not contain animals. Both the baseline (red) and CNN (blue) were set to yield a test set animal recall of 90%. As can be seen, both models produce false alarms, but the baseline predictions are scattered across all 1000×1000 tiles (dashed) compared to the single tile covered by the CNN.

further boost the precision, and *(iv.)* the introduction of a border class around animals that alleviates the confusion of nearby animals over background locations. Through an ablation study we demonstrated that all techniques are required together to reach the best performance.

In census applications, a pixel-precise localization of animals is only of secondary interest. To account for this fact, we proposed an alternative evaluation protocol that assesses model performances based on the number of detections they provide (compared to the actual number of targets), but poses less restrictions on the positional accuracy of predictions. For cases where animal detectors are used as a pre-selection step for a following manual verification stage, we further presented a second evaluation method that accounts for the number of image tiles a detector finds animals in. A good detector will only detect in the set of tiles that contain actual animals; a bottom baseline will require all tiles in the full dataset to be screened.

We demonstrated the effectiveness of all our recommendations in an experiment in the Namibian game reserve of Kuzikus, where we trained a CNN according to the presented recommendations and compared it to the state-of-the-art using the two evaluation protocols. The results show that the CNN not only manages to yield a substantially higher precision at high recall values, but also manages to produce more confined predictions, spreading across a lower number of image tiles. In practice, this results in the CNN lowering the number of image tiles to be verified to less than one third compared to the baseline, at a recall level of 90%. Our training recommendations are model-agnostic and straightforward to apply to any deep learning-based object detector, and the two evaluation protocols complement the model assessment. In sum, both parts demonstrate the necessity and a potential path to go towards animal censuses that are actually achievable during a realistic UAV campaign containing countless images with no wildlife recorded.

Acknowledgments

This work has been supported by the Swiss National Science Foundation (grant PZ00P2-136827 (DT, `http://p3.snf.ch/project-136827`). The authors would like to acknowledge the SAVMAP consortium (in particular Dr. Friedrich Reinhard of Kuzikus Wildlife Reserve, Namibia) and the QCRI and Micromappers (in particular Dr. Ferda Ofli and Ji Kim Lucas) for the support in the collection of ground truth data.

Chapter 3

Half a Percent of Labels is Enough: Efficient Domain Adaptation for UAV-based Animal Detection using Deep CNNs and Active Learning

This chapter is based on:

> **Kellenberger, B.**, Marcos, D., Lobry, S., and Tuia, D. (2019a). Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 57(12):9524–9533.

Abstract

We present an Active Learning (AL) strategy for re-using a deep Convolutional Neural Network (CNN)-based object detector on a new dataset. This is of particular interest for wildlife conservation: given a set of images acquired with an Unmanned Aerial Vehicle (UAV) and manually labeled ground truth, our goal is to train an animal detector that can be re-used for repeated acquisitions, *e.g.* in follow-up years. Domain shifts between datasets typically prevent such a direct model application. We thus propose to bridge this gap using AL and introduce a new criterion called *Transfer Sampling* (TS). TS uses Optimal Transport (OT) to find corresponding regions between the source and the target datasets in the space of CNN activations. The CNN scores in the source dataset are used to rank the samples according to their likelihood of being animals, and this ranking is transferred to the target dataset. Unlike conventional AL criteria that exploit model uncertainty, TS focuses on very confident samples, thus allowing a quick retrieval of true positives in the target dataset, where positives are typically extremely rare and difficult to find by visual inspection. We extend TS with a new window cropping strategy that further accelerates sample retrieval. Our experiments show that with both strategies combined, less than half a percent of oracle-provided labels are enough to find almost 80% of the animals in challenging sets of UAV images, beating all baselines by a margin.

3.1 Introduction

Repeated wildlife censuses provide an invaluable tool for ecologists to count animals, monitor population health and stem threats from poaching incidents (Hodgson et al., 2013; Yang et al., 2014). Population densities and spatial locations of big mammals are constantly fluctuating, and having up-to-date information on where and how many individuals are found may be decisive for grazing needs estimation, or for the success of anti-poaching means. Hence, authorities of national parks and game reserves require animal census tools that are fast, reliable, and suitable for repeated applications over time.

Traditional censuses using manual surveys from manned helicopters (Bayliss and Yeomans, 1989; Norton-Griffiths, 1978) are steadily replaced by approaches using UAVs (Linchant et al., 2015). UAVs are inexpensive, remotely-controlled aircrafts that can be equipped with small payloads like compact imaging cameras. Latest studies have shown censuses based on UAV imagery to yield superior accuracy compared to human surveys (Hodgson et al., 2018). They are especially appealing when combined with methods from machine learning and computer vision (Rey et al., 2017; Kellenberger et al., 2018c), in particular with object detectors (Ren et al., 2015; Redmon et al., 2016; Redmon and Farhadi, 2017) employing deep CNNs (Krizhevsky et al., 2012; LeCun et al., 2015): such models allow fast scans of the thousands of images UAVs produce over game reserves of average sizes (*i.e.*, hundreds of square kilometers), thereby alleviating the tedious work of manual photo-interpretation. This is particularly important in real-world scenarios where animals are a rare sight and images are dominated by empty background.

However, these models are typically trained on a single dataset and quickly break down in accuracy when applied to others. This problem is known as domain shift and denotes the inherent differences present between acquisitions (Tuia et al., 2016). For example, the image crops in Figure 3.1 are from the same game reserve, but are clearly very different in characteristics. For a human, it is trivial to locate the animals in either scene; a machine trained on only one set, however, is likely to fail when run on the other. In practice, even if it still finds most of the animals (high recall), such a model is likely to also produce false alarms everywhere in the background (low precision).

In the literature, this discrepancy is commonly solved by means of domain adaptation (Tuia et al., 2016), where a model trained on one dataset (*source* domain) is modified to also work on another (*target* domain). Multiple approaches have been proposed to this end, including unsupervised ones that only consider the images of the target domain, and semi-supervised methods that further assume the presence of a small number of labels (animal positions) in the target domain.

Figure 3.1: Examples from the Kuzikus dataset (see Section 3.3.1) from 2014 (left) and 2015 (right). It is trivial for humans to identify the animals in either image, but a model trained on only one dataset is likely to fail when predicting animals in the other.

As soon as the dominance and appearance variability of the background class get very high, a certain degree of supervision becomes unavoidable. This raises the question on how the few target labels can be obtained that are required to this end. A naive approach could require human operators to sift through hundreds of images before encountering an animal, which is highly inefficient and can lead to fatigue. This in turn likely causes erroneous labels, and hence missed targets and loss of accuracy. To this end, multiple studies have resorted to Active Learning (AL) (Tuia et al., 2011b). In AL, a machine (model) works hand-in-hand with a so-called oracle (typically a human expert) and exploits their knowledge by issuing queries for ground truth whenever it encounters a particularly relevant data sample.

The notion of relevance conventionally refers to the usefulness of a sample to the final model performance on the target dataset (Tuia et al., 2011b). Multiple AL criteria have been proposed (Settles, 2012): for example, uncertainty sampling methods like Breaking Ties (Luo et al., 2005) exploit the model's confidence on samples; model-specific approaches like margin sampling for Support Vector Machines (SVMs) (Schohn and Cohn, 2000), expected model change (Cai et al., 2013), or the recent Bayesian CNNs (Gal et al., 2017) make use of individual model properties to establish a sample ranking. They all seek for a prioritization of samples that lead to the highest performance of the underlying model with a small, given number of queries to the oracle.

In the case of animal censuses, however, things are different: instead of improved model generalization capability, park rangers are primarily interested in *locating the animals* in the new dataset. In this context, established criteria are likely to break down, since they tend to sample in areas where the detector is uncertain and the likelihood of obtaining a true positive is very low. Finding animals thus requires an AL criterion that works in the opposite direction by prioritizing predictions that are most certainly true positives (instead of low-confidence samples).

This means that two deviating objectives need to be met: fast animal localization on the one hand, but also some model improvement on the other. The latter refers to the interactive part of the AL adaptation paradigm: one could just train a detector on source, apply it once on the target images and then use a criterion in one go. This is known as "one-shot" AL (Santoro et al., 2016). However, we argue that using the newly obtained labels at every AL iteration to update the model and re-predict candidates can lead to increasingly higher quality predictions, and thus to higher chances of true positives retrieval.

In this chapter, we therefore present a novel strategy that allows finding as many animals as possible with minimal labeling effort in a new UAV acquisition, using an available source dataset and detector, AL and an oracle in the loop. In detail, the contributions of this work are as follows:

1. We introduce an AL criterion that, unlike conventional approaches, seeks to maximize the encounter probability of (rare) true positive candidates in the target domain.
2. Furthermore, we present a window cropping strategy that allows obtaining more labels per query while also being more intuitive for human annotators to label.
3. We provide an evaluation, comparison and ablation study on a UAV dataset of two distinct acquisitions characterized by domain shifts. Results show that, when using the proposed smart sampling strategy, it is possible to retrieve 80% of the animals by screening only half a percent of the acquired dataset.

The rest of this chapter is organized as follows:

Section 3.2 explains the main procedure, including the AL criterion denoted as "Transfer Sampling" (Section 3.2.1), as well as the window cropping strategy (Section 3.2.2). We put the model to the test in Section 3.3, results of which we show and discuss in Section 3.4. Finally, we draw conclusions from our work in Section 3.5.

3.2 Proposed Method

Figure 3.2 provides an overview of the proposed interactive domain adaptation workflow. As a precondition, it assumes the presence of a source dataset and an object detector (a deep CNN in our case) that has been trained on it. The model and its parameters are initially copied to extract features at every location in the images from the target domain. The distributions of these features in the source and target domain are then matched using OT (Cuturi, 2013), which allows transferring the source ground truth labels to the target domain. This provides a means of confidence prediction for the target samples, which can then be verified by an expert oracle.

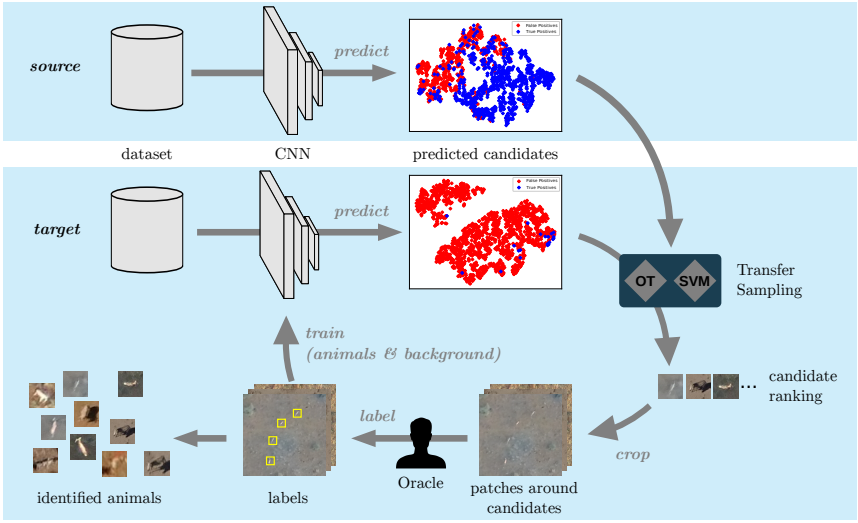


Figure 3.2: Overview of the proposed workflow. We first predict candidates in the source dataset using the original, unadapted CNN (top row). We do the same on the target dataset using the current CNN (below). These serve as inputs for our TS strategy (right), which ranks the source samples with an SVM and transports the ranking to the target candidates via OT. These ranked target candidates serve as anchor points for patches, which in turn form the query data to the oracle (bottom). The latter provides labels for subsequent CNN training (completion of the AL loop).

By iteratively querying the oracle to provide ground truth labels for the most likely true positives, the model can then gradually be fine-tuned to the target domain and provides better proposals in the following AL iterations, while making sure that the tedium of the prospective oracle is minimized. We limit experiments to ground truth-based (simulated) oracles in this work, but present a further strategy to accelerate and facilitate manual annotations in upcoming extensions with human annotators, which we denote as "window cropping." In the rest of this section, we discuss the two key components of the adaptation process: the proposed AL criterion for sample selection/ranking (Section 3.2.1), and the window cropping strategy (Section 3.2.2).

3.2.1 Transfer Sampling

As outlined above, our main interest lies in quickly locating animals in the target domain. Starting from the set of locations where the source model predicts more than 10%[1] chance of animal presence (denoted as *candidates* hereafter), we want to find those that are most likely to be true positives with the proposed AL criterion "Transfer Sampling" (TS).

In TS, we leverage the model's (higher) performance in the source domain and transfer this knowledge to the target samples. This is based on the assumption that the "best" predictions in source (*i.e.*, the true animals) are clustered together in the feature space of the last layer of the CNN, and that an equivalent region can be found in the target domain that is similarly relevant.

The challenge in this context is the imbalance between animals and background, combined with a likely excessive number of false detections made by the source model in the target dataset. To find the animals quickly and keep the annotators' motivation high, we thus need to prioritize target candidates whose corresponding source predictions were indeed true positives. We therefore consider sampling according to similarities in the CNN's feature space, spanned by the deep animal detector in the source and target domains. Figure 3.3 shows all samples in the source domain that were predicted by the source detector as "animals." Although the model still makes a number of false positives (red), a good majority of true positives (blue) is consistently clustered in one region of the feature space. In such a scenario, it therefore makes sense to start sampling in the far right blue area, since these are the furthest away from the transition to the false positives. We thus have two tasks to solve: *(i.)* numerically identify regions in the source domain feature space that most likely contain true animals, and *(ii.)* locate the same corresponding regions in the target domain.

For the first task, we resort to a margin-based auxiliary classifier to get a surrogate measure of sample certainty. In detail, we train an SVM (Cortes and Vapnik, 1995) on the full set of source candidates and then use it to rank the candidates by their distance to the separating hyperplane. This gives us an order that prioritizes samples as far away from the decision boundary as possible (the hyperplane distance is given as the marker size in Figure 3.3), which in turn makes sure that the most trustworthy candidates per source domain are sampled first. This strategy is conceptually close to margin sampling commonly used in AL (Schohn and Cohn, 2000), but we use it to focus the sampling on the *most* certain areas of the positive class, rather than the least certain ones.

The second task then consists of transferring the ranks to "similar" target samples. The intuition here is that both source and target candidates follow similar, mappable distributions, and we therefore need to find a way to establish an explicit source-to-target

[1] With 10% confidence we typically obtain recalls of 90% without having an excessive number of false positives.

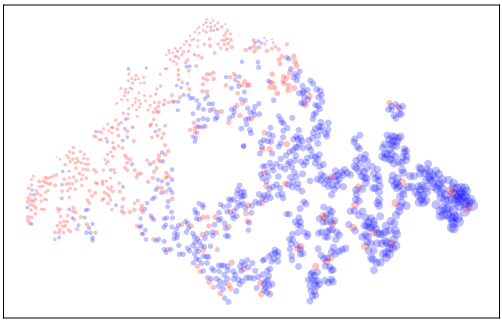


Figure 3.3: Source dataset candidates with animal confidence of 0.1 or more, projected using t-SNE (van der Maaten and Hinton, 2008). Blue samples were predicted correctly (true positives), red samples denote false alarms. The marker size indicates the distance to the SVM hyperplane (larger = further into the true positives region).

correspondence: given a predicted animal in the source domain, we want to know the predictions in the target domain that match to it. However, due to domain shifts a simple nearest neighbor search is likely to induce noise.

We propose to instead find this mapping using Optimal Transport (OT). OT finds a correspondence between two distributions that is optimal with respect to a global cost (Courty et al., 2017). It does so by calculating and minimizing for the Wasserstein distance, also known as the earth mover's distance. This distance quantifies the difference between the two distributions as a product of their data similarities and individual distances. The intuition behind this idea is that parts of the two distributions might be similar by some measure, but far apart with respect to their "location" within the distributions. In the case of discrete distributions like ours (*i.e.*, the distributions are constituted by individual predicted animal candidates), this means that two candidates from each distribution (resp. domain) only get associated with each other if they are similar by some measure *and* lie in similar areas of their respective distributions. In the following, we therefore assume the source and target domains to be represented by the discrete probability distributions $\mu_{\mathcal{S}}$ and $\mu_{\mathcal{T}}$:

$$\mu_{\mathcal{D}} = \sum_{i=1}^{n_{\mathcal{D}}} p_i^{\mathcal{D}} \delta_{\mathbf{z}_i^{\mathcal{D}}} \text{ for } \mathcal{D} \in \{\mathcal{S}, \mathcal{T}\}, \tag{3.1}$$

Here, the sum over all n locations (predicted candidates) of the domain $\mathcal{D}$, either source ($\mathcal{S}$) or target ($\mathcal{T}$), defines the discrete distribution. $\delta_i^{\mathcal{D}}$ denotes the Dirac at location $\mathbf{z}_i^{\mathcal{D}} \in \mathbb{R}^d$, with $\mathbf{z}_i^{\mathcal{D}}$ being the ith candidate's d-dimensional feature vector as predicted by

the CNN. $p_i^{\mathcal{D}}$ is the empirical probability per sample, to which we always assign the value $p_i^{\mathcal{D}} = 1/n_{\mathcal{D}}$.

This allows us to define the OT objective for the two discrete source and target distributions: to find a set of explicit links between all the individual source and target locations that match well. To this end, OT creates a sparse matrix γ of size $n_{\mathcal{S}} \times n_{\mathcal{T}}$, where $n_{\mathcal{S}}$ (resp. $n_{\mathcal{T}}$) is the number of samples in the source (resp. target) domain. γ contains non-zero values wherever specific source and target locations "match." This match is defined as the link contributing to a global cost $\mathbf{C}$ for the two samples being minimal. Intuitively, establishing a link between a source and a target sample that both lie in similar regions in the feature space induces a lower cost than if they were *e.g.* in opposite regions. Numerically, the optimal solution to that, *i.e.* the optimal *transport plan*, can be obtained as follows:

$$\gamma^* = OT(\mu_{\mathcal{S}}, \mu_{\mathcal{T}}) = \arg\min_{\gamma \in \mathcal{B}} \langle \gamma, \mathbf{C} \rangle_F, \tag{3.2}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product and $\mathbf{C}$ the cost matrix of size $n_{\mathcal{S}} \times n_{\mathcal{T}}$. $\mathbf{C}_{ij}$ is the cost to move a unit amount from $z_i^{\mathcal{S}}$ to $z_j^{\mathcal{T}}$ (at source and target locations i and j, respectively). $\mathcal{B}$ is the so-called transportation polytope, *i.e.* the set of all possible, positive matrices with prescribed marginals $\mu_{\mathcal{S}}$ and $\mu_{\mathcal{T}}$. In other words, $\mathcal{B}$ comprises all combinations of transport links between all $n_{\mathcal{S}}$ source and $n_{\mathcal{T}}$ target samples. For the cost term $\mathbf{C}$, a commonly used choice is the ℓ_2 norm between samples (Courty et al., 2017). We follow this approach, as we found it to work well in our setting involving CNN features. Equation (3.2) can be formulated as a linear program, and further be solved efficiently with simplex-based algorithms as well as group regularizations (Cuturi, 2013; Courty et al., 2017). This then gives us the optimal transport plan $\gamma^* \in \mathbb{R}^{n_{\mathcal{S}} \times n_{\mathcal{T}}}$, which provides explicit correspondences between individual source and target samples that are sound with respect to the whole distributions. We note that in general, and specifically when $n_{\mathcal{T}} > n_{\mathcal{S}}$, the coupling may yield one-to-many linkages, and many-to-one in the inverse case, but is always sparse thanks to the constraint that the source and target marginal probabilities (Equation (3.1)) must sum to one.

We can now use this transport plan to transfer the SVM-derived source scores to the individual target samples:

$$s_j^{\mathcal{T}} = \frac{1}{N} \sum_{i=1}^{n_{\mathcal{S}}} s_i^{\mathcal{S}} \delta(\gamma_{ij} > 0) \tag{3.3}$$

Here, $s_i^{\mathcal{S}}$ denotes the distance to the SVM hyperplane for the ith source sample and $\delta(\cdot)$ is the Kronecker delta, returning value 1 if the condition inside the brackets is true, and 0 otherwise. $s_j^{\mathcal{T}}$ is the score for the jth target sample. In essence, we assign a score to each target sample as the sum of the SVM hyperplane distances of those *source* samples whose OT link (γ_{ij}) is non-zero, normalized by $N = \sum_{i=1}^{n_{\mathcal{S}}} \delta(\gamma_{ij} > 0)$. An exemplar mapping on

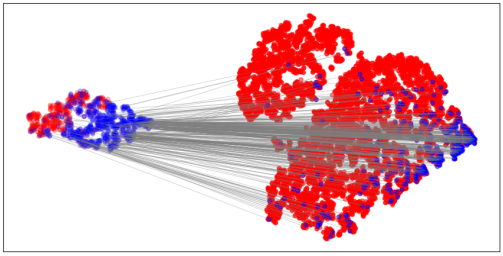


Figure 3.4: A subset of predicted locations in the source (left) and target (right) domain training sets. Blue samples were predicted correctly (true positives) and red samples are false positives. Gray lines denote the correspondences found by OT for all the correctly predicted target samples. Note that, despite the imbalance and higher number of false alarms in the target set, the OT correspondences are globally consistent.

a subset of the training data is shown in Figure 3.4. This figure shows samples predicted by the CNN that has been trained on the source (left point cloud), but not yet adapted to the target domain (right point cloud). The gray lines show links obtained by OT[2]. At a first glance, it is evident that the CNN predicts orders of magnitude more false positives in the target domain, which is due to the domain shift between the two datasets. If we follow the OT links from all *source* true positives, we hit 51 target true positives (around 10% of the target true positives) and 759 target false positives (around 2.6% of the target false positives). This may sound like a low-precision result, but note that we prioritize the source true positives with our TS metric, thus drastically reducing the number of false positives (see results below). Also, the OT links to the true positive target samples consistently come from true positive source samples, which indicates that the OT-derived transport plan is globally sound and succeeds in mapping correct predictions together. In the end this means that TS is particularly robust to class imbalances: even if the ratios of true to false positives differ substantially between the two domains (as is the case in our study), TS still prioritizes the most confident predictions. Once the costs are transported to the target domain, we only need to rank the target samples and can further sample them with priority on high-quality predictions.

3.2.2 Window cropping for patch-based labeling

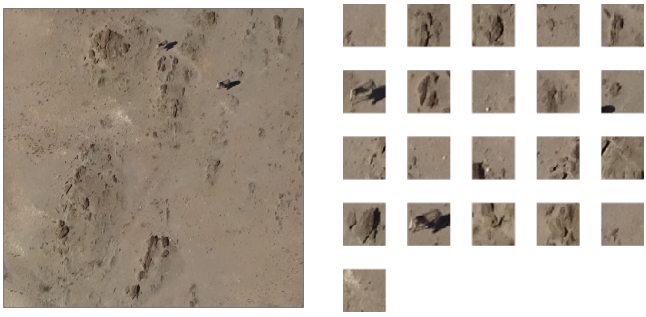
The second major component of our model, the window cropping strategy, extends the queried candidate with its spatial surroundings. In other words, for all candidates iden-

[2]For illustration purposes we only show links that point to true positives in the target domain.

tified through TS, we crop a patch of fixed size around them and have the entire area labeled by the oracle, instead of the single prediction.

As mentioned above, we seek to find a trade-off between simply locating animals and CNN updates. Window cropping enhances both objectives, as it increases the total amount of labels obtainable from the oracle in a single query. We can crop a window around a query position in a UAV image in such a way that it includes as many other predicted candidates as possible. In the case of false positives, this increases the information flow to the CNN and results in it making less false predictions during the next AL iteration. However, in case neighboring candidates are also true positives, window cropping can accelerate the retrieval rate of animals with minimal additional effort from the oracle. This is not unlikely, since animals tend to flock together in groups. If the CNN thus finds just one of the animals in a herd, it is trivial for humans to localize the rest close-by this way.

A further advantage of including the neighborhood lies in the ability of humans to be able to instantly recognize targets, if spatial context is provided. Consider Figures 3.5a (sample target image) and 3.5b (predicted candidates in it): querying the oracle for every candidate individually would not only be too exhaustive, but also more difficult, as the recognizability of the target depends heavily on spatial context, which might be missing (or confusing) on a per-sample query. In turn, locating animals in an adequately sized patch is a much simpler task for humans.



(a) **(b)**

Figure 3.5: Patch of a target image (left) and all candidates predicted by the source CNN in it (right). By cropping a larger patch around multiple candidates at once (left), the labeling process is both faster and more intuitive for human operators than querying on a per-candidate basis (right).

It thus makes sense to crop patches in such a way that they include as many neighboring candidates as possible. We first define a patch rectangle as $r = \{r_x, r_y, r_w, r_h\}$, with r_x and r_y denoting the top-left corner of it, and r_w and r_h the width and height in pixels, respectively. Also, let $l = \{l_x, l_y\}$ be the position in the image of the candidate selected

by the AL criterion, further referred to as the *anchor point*. To select the best rectangle around the anchor point, we optimize a function that maximizes the number of candidates in the patch, minimizes the overlap with previously cropped patches and keeps the current candidate as close as possible to the center of the patch window:

$$r^* = \underset{r \in \mathcal{R}_l}{\arg\min} \left((1 - N(p, r)) + \max(I(r, \mathcal{R}_q)) + \lambda \left\| r_c - l \right\|^2 \right) \tag{3.4}$$

where $\mathcal{R}_l$ is the set of windows that contain the anchor point l. The first term, $N(p, r)$, is the number of candidates p inside rectangle r, normalized by the total number of candidates present in the image. The second term, $\max(I(r, \mathcal{R}_q))$, denotes the maximum area intersection between rectangle r and all the rectangles in this image that have been queried before ($\mathcal{R}_q$). This term is normalized by the area of the rectangle so that it also sums to one, like the first term. The third term compares the anchor point l with $r_c = \{r_x + r_w/2, r_y + r_h/2\}$, *i.e.* the center of the rectangle, by means of a norm and favors centering the window on the anchor. This last term primarily plays a role when the image only contains the anchor point l (*i.e.*, there are no other candidates nor any previously queried rectangles); hence, it is downweighted with a constant λ (set to 0.01 in the experiments). Example scenarios for the three terms are shown in Figure 3.6. This score function is non-differentiable in multiple ways. However, since we restrict r to always contain anchor point l, the search space $\mathcal{R}_l$ is very limited. We thus are able to employ an exhaustive grid search around the anchor point.

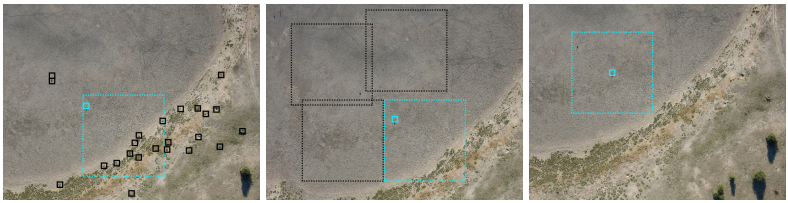


Figure 3.6: For window cropping, we address different scenarios to maximize the query gain: in the first situation (left), we place the candidate window (cyan dotted) so that it includes the anchor point selected by the querying strategy (cyan solid) and as many other candidates (black) as possible. In the second situation (middle), we minimize the overlap with previously queried windows (black dotted). If neither other candidates nor previous windows are present (right), we position the window centered around the anchor point.

We then query the oracle with this patch and receive positions of animals within, if present. Any other location in the image is labeled as background.

This procedure naturally depends on the patch size, where a compromise must be found: too large patches make it increasingly harder for humans to label, while too small patches

exhaust the querying budget too quickly and provide less context. In this study, we limit experiments to a simulated oracle, but nevertheless use a patch size that we found reasonable while manually labeling the dataset. In detail, we crop patches of 1000×1000 pixels (approx. 60×60m) that provide a sufficiently large number of samples while still being easy for humans to label.

3.3 Experiments

We now put the proposed workflow to the test and describe the data and parameters below. Section 3.3.1 describes the two datasets used; Section 3.3.2 highlights parameters of the detector CNN and of the AL routine.

3.3.1 Study Area and Data

We evaluate our proposed method on UAV datasets acquired over the Kuzikus wildlife reserve in Namibia[3]. Kuzikus is a private-owned park in the African savanna and home to multiple species of large mammals like kudus, giraffes, zebras, black rhinos, and more. In total, more than 3000 individuals are spread across an area of 103km^2 (Rey et al., 2017; Kellenberger et al., 2018c).

In 2014 and 2015, two image acquisition campaigns were carried out by the SAVMAP consortium[4]. A SenseFly eBee[5], equipped with a consumer-grade RGB compact digital camera, was employed for both campaigns. This resulted in 654 images for the 2014 campaign, and 3254 for the year 2015. The images of the first acquisition were initially labeled in a crowd-sourcing operation organized by MicroMappers[6] (Ofli et al., 2016), followed by several iterations of refinement by the authors. The 2015 images were completely labeled by the authors.

The final statistics for both datasets are listed in Table 3.1. Although both datasets were acquired over geographically overlapping areas, they feature a substantial domain shift in multiple ways: in terms of *external conditions*, the datasets were acquired at different times of the year (May 2014, resp. February and May 2015), under different weather and lighting conditions, with different cameras and varying flying altitudes above ground. Furthermore, additional shifts can be observed in the *label space*: the 2014 data already have a substantial class imbalance (1:10'000 in terms of animal-to-background pixels), but the 2015 dataset is larger and even more imbalanced, with an overall lower proportion of animals.

[3] `http://kuzikus-namibia.de/xe_index.html`
[4] `https://lasig.epfl.ch/savmap`
[5] `https://www.sensefly.com`
[6] `https://micromappers.wordpress.com`

Table 3.1: Overview of the 2014 and 2015 Kuzikus UAV datasets.

	Set 1	Set 2
Year	2014	2015
Image sizes	4000×3000	4896×3672, 4608×3456
Camera models	Canon PowerShot S110	Sony DSC-WX220, Canon IXUS 127 HS
No. images	654	3254
↪ with animals	239	111
↪ without	415	3143
No. animals	1183	646
Elevation a.g. (est.)	120m	160m

We divided the source (2014) dataset into training, validation and test splits according to the following set of rules:

- We assign entire images to only one of the three sets to avoid autocorrelation effects.
- We differentiate between images that contain at least one animal and empty ones. All the images with at least one animal are distributed so that the number of *animals* in the sets are distributed as follows: 70% for the training set, 10% for validation, and 20% for testing.
- All the remaining images (*i.e.*, those without any animal) are then distributed at random to meet the same 70-10-20 split, but this time based on the number of *images*, as closely as possible.

For the target (2015) data, we do not require a test set, since we sample directly using the oracle. However, we do use a small validation set for hyperparameter fine-tuning. Details on the dataset splits can be found in Table 3.2.

3.3.2 Model Setup

In the following, we highlight the main model components and their parameters: Section 3.3.2 explains the deep CNN used for animal detection, and Section 3.3.2 provides details on the AL framework.

CNN Training

In this study, we follow the training recommendations presented in Kellenberger et al. (2018c), which are specifically tailored to animal censuses in heavily imbalanced datasets.

Table 3.2: Split properties for the 2014 and 2015 datasets.

			2014	2015
Training	No. images	with animals	159	91
		without	291	2750
		total	450	2841
	No. animals		830	565
Validation	No. images	with animals	35	20
		without	41	393
		total	76	413
	No. animals		118	81
Test	No. images	with animals	45	-
		without	83	-
		total	128	-
	No. animals		235	-

These recommendations include class weights; a special "border" class that is placed in the 8-neighborhood of an animal location to reduce multiple detections of it; curriculum learning, where the model is first trained on images that always contain an animal; and hard negative mining, which amplifies the weights of the four most confidently predicted false alarms after epoch 80.

We further adopt the detector CNN from Kellenberger et al. (2018c), whose architecture is shown in Figure 2.8. The model accepts image patches of size 512×512 and predicts a downsampled grid of animal probabilities (32×32). In addition to the basic architecture presented in Chapter 2, we replace all batch normalization layers with instance normalization (Ulyanov and Vedaldi, 2016), which performs unit-norm scaling for each image in the respective mini-batch individually. We found this substitution not to harm the model performance, but to help stabilize prediction consistency by avoiding dependencies on mini-batch configurations both during training and testing.

For the fine-tuning stages throughout the AL iterations, we lower the learning rate from 10^{-6} (used on source) to 10^{-7}—we found this to prevent oscillation effects on the reduced-sized target training set. Also, we disable curriculum learning and train on the full growing AL-derived dataset at every iteration. All other parameters and training procedures, such as hard negative mining, are kept the same as in the source model.

AL Loop

In a pre-stage, we train our detector CNN until convergence on source. We use this CNN to obtain all candidates in the source training set, keeping all predictions with animal confidence of 0.1 or higher. To reduce the number of double-predictions, we employ Non-

Maximum Suppression (NMS) with a search radius of 2 prediction grid cells, retaining only candidates in a 4-neighborhood with the highest confidence.

Next, we run a total of ten AL loops, querying 50 patches of 1000×1000 pixels size per iteration. We believe 50 patches to be a fair amount to query without risking the human annotators to make errors due to fatigue (note that other studies used query sizes of up to 200 images in more heterogeneous datasets (Kao et al., 2018)). Using those we fine-tune the CNN on the target training set for 12 epochs per AL iteration, which we deem a reasonable trade-off between training time and accuracy gain. Since we queried patches that are larger than what the CNN accepts as an input, we can perform extra data augmentation by randomly cropping sub-patches of the labeled areas.

We compare our TS strategy to three baselines: random candidate selection, Breaking Ties (Luo et al., 2005), and CNN confidence for being an animal ("max confidence"), all with window cropping enabled. Since model fine-tuning and candidate re-prediction is computationally intensive, we assess two scenarios for each sampling strategy: one with CNN fine-tuning at every AL iteration, and one without (*i.e.*, using only the source model to predict candidates once and querying with TS only on those samples).

3.4 Results and Discussion

Figure 3.7 shows the number of animals found over the course of the ten AL iterations. Already after the first 50 queries, TS found 122 animals and is significantly ahead of the baselines. This trend continues throughout the iterations, and after five AL iterations, TS found 443 out of the total 565 animals (78.4%). At this stage, the oracle had been queried 250 times. Afterwards, the total number of correctly identified animals slightly rises to 473 (83.72%). Our window cropping algorithm allows sampling patches in an out-of-grid fashion. However, if we assume uniform sampling on a grid per image, the target training set would consist of 54'324 queryable patches. This means that TS only requires the user to review around half a percent of the dataset in order to find almost 80% of the animals.

In comparison to all baselines, TS manages to yield a higher recall almost throughout the entire process. Although the max confidence ranking manages to reach roughly the same level, it does so only after the sixth AL iteration. TS in turn identified the same number of animals already an entire iteration earlier, and stayed above the rest until then by quite a margin. This means that substantially less queries need to be made to the annotator when using TS, resulting in faster convergence and hence a more economical retrieval process.

Figure 3.8 shows the selected patches, predicted candidates and ground truth for both TS (top image) and max confidence (bottom): the latter does manage to find a reasonable

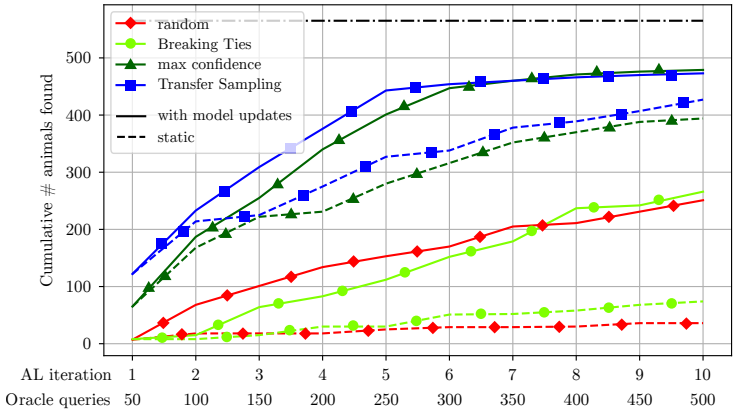


Figure 3.7: Cumulative number of animals found over the AL iterations. Solid lines denote the criteria performances with model updates and target candidate re-predictions at every iteration; dashed lines are the static performances (continuous sampling on the initially predicted candidates). The black dash-dotted line marks the total number of animals in the target training set. Best viewed in color.

number of true positives, but nonetheless misses more than half of the animals present in the scene. Explanation may be found in the t-SNE plots (right side of each image), which indicate that most of the true positives are to be found primarily in one area in the bottom right of the feature space. One might expect a correlation between feature space locations and confidences, but as shown here, this is only partially the case: TS manages to get ahead of max confidence by strictly sampling in this area of high true positives concentration (bottom right of t-SNE plots), instead of according to confidences.

The performance of the other baselines is less satisfactory. For instance, the focus of Breaking Ties lies on minimizing the model's uncertainty. It is therefore unsurprising that it fails even worse than random sampling in finding animals. However it is still able to locate some of the animals. Consider Figure 3.9, which shows selected patches and predicted candidates on an example for all strategies. In this case, random sampling managed to find one of the animals in an earlier iteration, but misses the rest. Breaking Ties sampled two times in the image and found three animals, but this was most likely due to neighboring candidates that do not represent true positives (note the candidates predicted by Breaking Ties lying all on transition areas from shadow to ground). Max confidence again found the same three animals, but missed the hotspot in the center-right portion of the figure. Lastly, only TS managed to locate every single animal present, and did so with a minimal number of queries. In essence, the other strategies occasionally show true positives, but mainly due to window cropping.

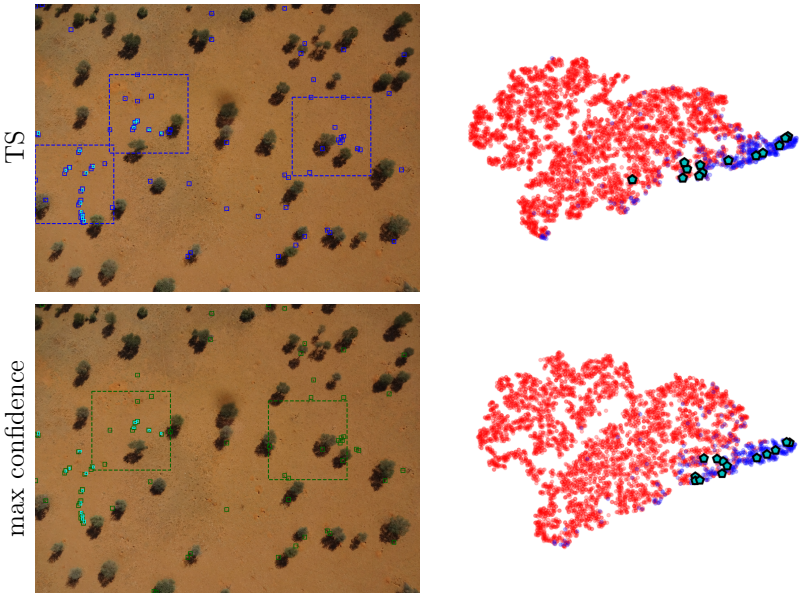


Figure 3.8: Prediction examples for the TS (top) and max confidence (bottom) strategies after five AL iterations (250 queries), together with their respective t-SNE plots of the predicted candidates (right). Cyan pentagons in the t-SNE plot correspond to the cyan ground truth bounding boxes in the UAV images.

Finally, in all cases we observe a significant boost when the model is updated and candidates are re-predicted versus the static "one-shot" prediction and sampling (Figure 3.7). This indicates that the labels provided by the oracle at each stage are useful and complete enough for adapting the CNN to the target domain, so that the candidates predicted during the next iteration are of higher quality. Window cropping helps particularly in this case, since CNNs require a large number of training samples: on the one hand, it increases the amount of image data seen by the model. On the other, it also increases the information gain: since the cropping strategy maximizes the number of predicted candidates to be included per patch, it maximizes both the number of true and false positives in the new dataset. In this respect both true and false positives are vital for the model, improving its abilities to better separate animals from background in the target dataset.

Noteworthy in this context is the difference in the number of predicted candidates between TS and the other strategies. After the fifth AL iteration, the CNN trained with TS produced a significantly higher number of candidates than all the baselines, which can be seen by the high number of blue squares in Figure 3.8. A possible explanation for this

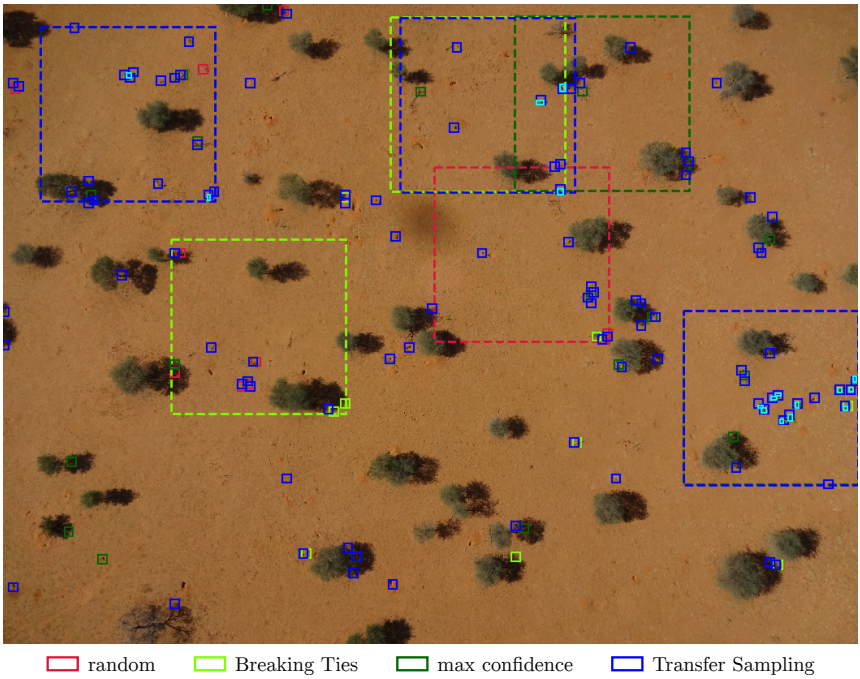


Figure 3.9: Example image from the target training set with annotations after five AL iterations, showing all selected patches per sampling strategy (dashed), the candidates predicted by each CNN at the very last AL iteration, as well as the ground truth (cyan rectangles). Since only the last predictions, but all patch rectangles are shown, some of them (*i.e.*, from earlier AL iterations) do not appear to encompass any candidate.

phenomenon is that TS finds more animals already in the first AL iteration: the CNN is then fine-tuned with a lot more true positives and therefore predicts more candidates when re-applied to the target dataset. In a fully-automated evaluation setting (*i.e.*, without any oracle input), this could be problematic, since this increased number of predictions also results in more false positives. However, thanks to TS we can filter the predictions very efficiently, and as a result, localize the animals even in increasingly imbalanced settings.

3.5 Conclusion

In this chapter, we have studied the task of repeated animal censuses on UAV imagery by addressing the domain adaptation problem involved. To do so, we integrated a deep CNN-based animal detector in an AL loop. The core component of our strategy is the AL criterion: unlike traditional approaches that seek to maximize the model performance on the new dataset, our *Transfer Sampling* (TS) criterion is designed to localize the rare animals in tens of thousands of false alarms as efficiently as possible. TS works by leveraging the superior performance of the CNN detector in the source dataset (which it had been trained on) and transferring this knowledge to the target set using the distribution-mapping framework Optimal Transport. The number of hits was further raised by integrating a smart window cropping strategy that maximizes the number of detections to be labeled per query, while making the labeling process itself more intuitive. Our experiments in the Namibian natural reserve Kuzikus have shown that TS indeed outperforms other AL criteria by a large margin and allows retrieving 78.4% of the animals in just 250 queries, resp. by having the oracle review less than half a percent of the entire dataset. In effect, this method thus allows for efficient and economic repetitions of animal censuses as it integrates both the adaptation and required manual verification stages into one optimized, interactive workflow.

Future work may extend this concept in multiple ways: for example, experiments with human annotators instead of simulated oracles would highlight requirements by park rangers. Extending TS *e.g.* by a measure of the user's confidence in providing a ground truth (Tuia and Muñoz-Marí, 2013) could improve the real-world applicability of such a system. On a different track, adaptations to other geographical areas instead of new acquisitions over the same game reserve would allow testing the strategy under potentially even stronger domain shifts.

Acknowledgments

This work has been supported by the Swiss National Science Foundation grant PZ00P2-136827 (DT, `http://p3.snf.ch/project-136827`). The authors would like to acknowledge the SAVMAP consortium (in particular Dr. Friedrich Reinhard of Kuzikus Wildlife Reserve, Namibia) and the QCRI and Micromappers (in particular Dr. Ferda Ofli and Ji Kim Lucas) for the support in the collection of ground truth data. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan V GPU used for this research.

Chapter 4

Weakly-supervised Wildlife Detection in UAV Images

This chapter is based on:

> **Kellenberger, B.**, Marcos, D., and Tuia, D. (2019b). When a few clicks make all the difference: improving weakly-supervised wildlife detection in UAV images. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRw)*.

Abstract

Automated object detectors on Unmanned Aerial Vehicles (UAVs) are increasingly employed for a wide range of tasks. However, to be accurate in their specific task they need expensive ground truth in the form of bounding boxes or positional information. Weakly-Supervised Object Detection (WSOD) overcomes this hindrance by localizing objects with only image-level labels that are faster and cheaper to obtain, but is not on par with fully-supervised models in terms of performance. In this study we propose to combine both approaches in a model that is principally apt for WSOD, but receives full position ground truth for a small number of images. Experiments show that with just 1% of densely annotated images, but simple image-level counts as remaining ground truth, we effectively match the performance of fully-supervised models on a challenging dataset with scarcely occurring wildlife on UAV images from the African savanna. As a result, with a very limited amount of precise annotations our model can be trained with ground truth that is orders of magnitude cheaper and faster to obtain while still providing the same detection performance.

4.1 Introduction

Object detection in images from UAVs recently experienced an uprising interest in the computer vision community (Bondi et al., 2018a; Du et al., 2018; Bazi and Melgani, 2018; Ofli et al., 2016; Attari et al., 2016). Applications are manifold and range from security and safety (Idrees et al., 2018) to animal conservation (Arteta et al., 2016; Bondi et al., 2018a). Thanks to research advancements like Convolutional Neural Networks (CNNs) (LeCun et al., 2015), automated detectors have shown significant increase in positional and classification accuracy of objects (Ren et al., 2015; He and Girshick, 2017; Redmon et al., 2016; Redmon and Farhadi, 2017).

Traditionally, object detectors are fully-supervised, which implies that the class, location in the image, and dimensions of every object of interest is known during training. This high level of supervision leads to superior model performances, but comes at a substantial annotation cost: it requires precise bounding boxes to be drawn, and class labels to be given, to many objects in hundreds or thousands of images (as in Figure 4.1). This becomes especially tedious for datasets that contain large numbers of objects per image, such as crowd surveillance imagery (Wang et al., 2018), as well as cases where objects of interest are a rare sight. UAV images bear particularly high implications for ground truth cost and quality in this respect (Ofli et al., 2016).

A promising direction to ease label complexity is WSOD, where a model is only given image-level labels, but aims at localizing objects in the image nonetheless (Bilen et al., 2015). This concept is also related to density estimation (Wang et al., 2018) in that in both cases no bounding boxes or object positions are provided as a ground truth. Such models require much simpler and therefore cheaper annotations, such as image-level labels in the WSOD case, and counts (number of class instance occurrences) in density estimation. Consider Figure 4.1: drawing bounding boxes for all objects in the UAV images would be tedious and hence expensive, especially when thousands of such images need to be labeled. In turn, providing object count estimates or even just the presence or absence of objects for the entire images can be done with just one interaction per image. The downside to WSOD models is typically a loss in accuracy: such models tend to miss small objects (Oquab et al., 2015) or combine multiple close-by instances of the same class into one (Gao et al., 2018). Count-based models need to be trained with a sufficient number and variation of objects per image, sometimes including images that contain no object at all, and they risk to focus on the wrong kind of object if two classes appear in similar numbers in the imagery, a scenario not unlikely in UAV images. For example, one may be interested in the number of cars in a top-down view image, but the detector could get confused due to similar numbers of motorcycles.

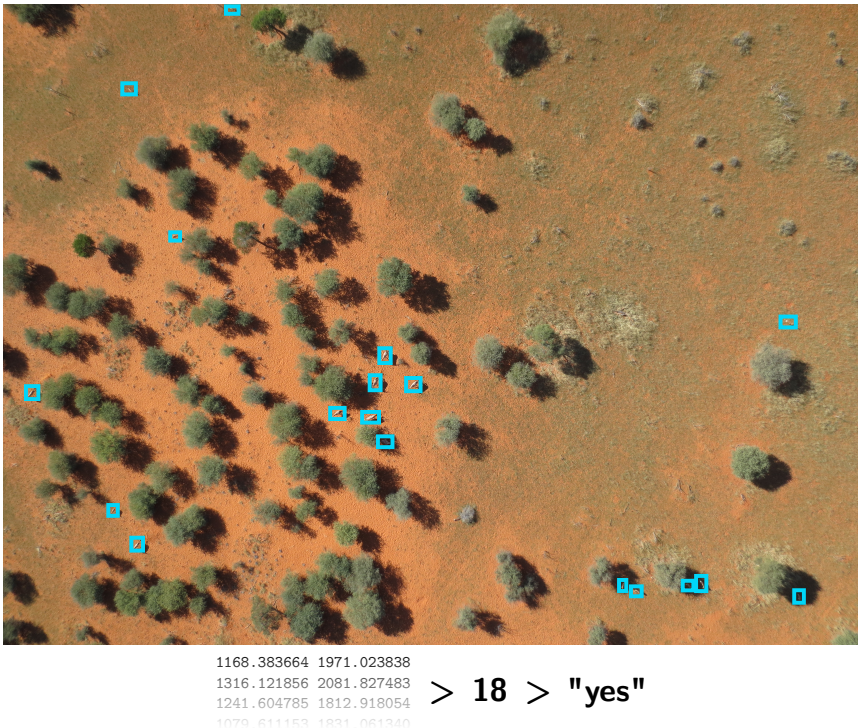


Figure 4.1: Large-scale UAV datasets pose substantial labeling efforts if bounding boxes or object coordinates (bottom left) are required. Our model reduces this tedium by resorting to inexpensive, image-level object counts (bottom middle) or simple object presence/absence (bottom right) for the most part, and requiring positions only for a handful of images.

In this chapter we propose to overcome these issues by combining the merits of both fully- and weakly-supervised detectors in a hybrid approach, as shown in Figure 4.2. In detail, we train a WSOD model with weak supervision (object count or binary presence/absence of objects) and then complement it with a small fraction of images where a full, positional ground truth is available. The intuition behind this is that few spatial ground truth maps are comparably inexpensive to obtain, but sufficient for the model to focus on the object class of interest, so that predictions become more precise and ambiguities are reduced. We evaluate our models on a challenging, UAV-derived dataset and show that, even with completely weak supervision, object localization is possible to a high accuracy, but that adding just 1% of full positional ground truth can match the performance of a model that has profited from 100% full supervision.

4.2 Related Work

4.2.1 Object counting

Object counting models recently gained a lot of attraction in the computer vision community (Liu et al., 2017; Kang et al., 2018; Sindagi and Patel, 2018; Zhang et al., 2015). They generally fall into one of three categories: *detection-based* models (Ren et al., 2015; Redmon et al., 2016; Redmon and Farhadi, 2017; Girshick, 2015) first localize objects of a kind in an image and then simply return the number of detection. These models provide explicit localizations, but thus require large numbers of expensive bounding boxes. Also, they can be strongly affected by cases of occlusions and pose variations, since each object instance is handled as a binary contribution to the total count. *Regression-based* models (Shang et al., 2016; Ryan et al., 2015) directly predict an estimated number of objects from the image. They forgo expensive ground truth, but may struggle whenever they have to do extrapolation (*i.e.*, predicting more or less objects than seen in the training set) and cannot easily provide object positions. Finally, *density-based* models (Kang et al., 2018; Zhang et al., 2016) predict a spatial heat map of localizations whose sum corresponds to the expected number of objects. These models basically combine the advantages of both detection- and regression-based approaches in that they only need count estimations but still yield a spatial prediction. If successful, these models can therefore be seen as a variant of WSOD, which is discussed below.

4.2.2 Weakly-Supervised Object Detection

The aim of WSOD is to localize objects in images with just image-level ground truth. The hope of such models is that they will learn which parts of the image to draw their attention to, and that these parts will then coincide with the location of the objects of interest. Several variants of WSOD models have been proposed: Bilen et al. (2015) retrieve reoccurring patterns across the dataset by clustering. Papadopoulos et al. (2016) train an object detector through active learning, querying an oracle for automatically predicted bounding boxes. Shi and Ferrari (2016); Shi et al. (2017) both exploit prior knowledge (object sizes, texture compositions of objects) to facilitate identification of objects across images. The work of Oquab et al. (2015) is closely related to heatmap-based models used in this study, but differs in that the model is only trained with weak supervision, and that the prediction is a grid of pseudo-probabilities that gets softmax-activated across all classes. The authors report good localization performance for vision benchmark datasets, but due to the softmax operator encouraging weight concentration to one sample, their model is likely to be unable to localize multiple objects of the same category in one image, a crucial necessity for counting. Gao et al. (2018) attempts to solve this drawback by disentangling multiple objects detected together, based on a count-based ground truth,

but their method still relies on explicit detectors that may be hard to train and may fail under occlusions and on small targets.

4.2.3 Hybrid Approaches

Some works studied combining different levels of supervision, but typically focus on performance improvements, rather than on reducing annotation efforts. Idrees et al. (2018) employ both positions and counts for crowd density estimation, but needed a dataset with complete positional information for over 1.5 Million targets. Liu et al. (2017) propose a model that uses detection for low estimated object counts and switches to regression for crowded scenes. Papadopoulos et al. (2017b) detect objects with image-level labels and single point annotations; Bearman et al. (2016) extend this idea to semantic image segmentation. However, both approaches require all of the training images to be labeled this way. The work of Zhang et al. (2015) is more similar to our model in that the authors also alternate between weak supervision (density) and object position ground truth for training, but also their scheme requires all training set instances to be fully labeled.

4.3 Method

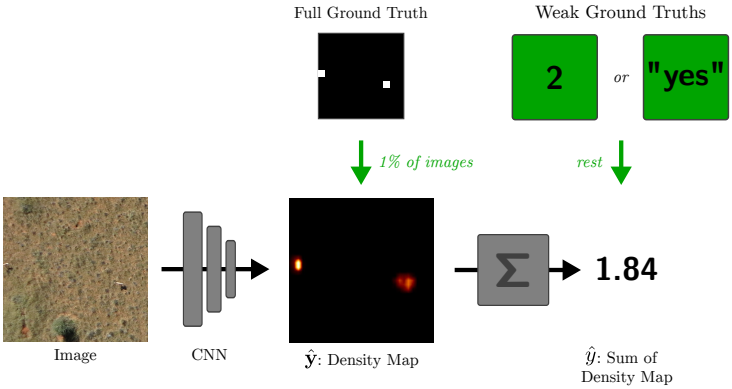


Figure 4.2: Overview of our proposed model. The detector predicts object locations via a heat map and receives weak supervision, either through object counts or presence/absence of objects as a ground truth. In a small fraction of cases (*e.g.* 1%), we train on the full, positional ground truth to improve performances.

4.3.1 Density-based Object Detection

As a base model for all experiments we employ a deep fully-convolutional CNN that accepts an image and predicts a downsampled grid $\hat{\mathbf{y}}$ of size $N \times M$. This grid corresponds to the density map and is expected to contain high values in those locations where an instance of our object class of interest is to be found, and values close to zero elsewhere. A non-negative activation nonlinearity, such as a Rectified Linear Unit (ReLU) or sigmoid, ensures that predictions are bound to a range of zero or more. Let scalar $\hat{y} = \sum_{i \sim N} \sum_{j \sim M} \left(\mathbf{y}_{ij}\right)$ be the sum over the output grid $\hat{\mathbf{y}}$ for each image. Our objective then is it to train the model to predict a grid whose sum $\hat{y}$ corresponds to the ground truth. In this chapter we consider two levels of weak supervision: *(i.)* a scalar for the object count; *(ii.)* a binary variable denoting presence/absence of the object class of interest in the image. The latter is an extreme case that we use to see if a CNN can localize objects where the only knowledge we have is whether the class of interest is present in the image, eliminating the need for the annotator to count the objects to create the ground truth and thus greatly reducing the annotation effort.

To this end we employ different loss functions, depending on the level of weak supervision. In cases where the ground truth consists of an object count, we employ a Smooth $\ell 1$, also known as Huber loss (Girshick, 2015):

$$\ell(y, \hat{y}) = \ell_{Huber}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |y - \hat{y}| < \delta \\ |y - \hat{y}| - 0.5\delta & \text{otherwise} \end{cases} \tag{4.1}$$

where y is the object count in the image. Unlike an $\ell 2$ loss, the Huber loss employs an $\ell 1$ norm beyond a threshold δ (set to 1 in our experiments), which poses less of a restriction on outliers in the data. This is also useful for balancing loss strengths between samples with low and high counts, which is the case in our UAV images (wildlife tends to flock together, causing some images to have high counts and others to contain no animal at all).

In the binary case, we only have two possible labels: 0 (absence) and 1 (presence). We therefore use a conditional loss instead:

$$\ell(y, \hat{y}) = \begin{cases} 0 & \text{if } y = 1 \text{ and } \hat{y} \geq 1 \\ \ell_{Huber}(y, \hat{y}) & \text{otherwise} \end{cases} \tag{4.2}$$

In other words, the loss here pushes the sum over the predicted heat map to zero whenever no object is present ($y = 0$), but is zero if the respective area is labeled as "has object" ($y = 1$) and the sum over the heat map is at least one.

Note that Equations (4.1) and (4.2) are limited to a single object class, as is the case in our experiments, but an extension to the multi-class scenario would be straight-forward

by predicting K heat maps for K classes and applying the respective loss function individually.

4.3.2 Sparse Full Supervision

Our proposed method follows the same principle and employs the same loss functions for the most part. For the weakly labeled images the training loss therefore corresponds to either Equation (4.1) or (4.2), depending on whether a count- or presence/absence-based ground truth is available. However, for a small percentage of images we assume the availability of precise, positional ground truth for all objects they contain (left ground truth in Figure 4.2). For these images we replace the weakly-supervised loss with a spatially explicit binary cross-entropy loss:

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i \sim N} \sum_{j \sim M} -\big(y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})\big) \tag{4.3}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}} \in \mathcal{R}^{N \times M}$ are the ground truth and prediction (density) maps, respectively. $\mathbf{y}$ contains value one wherever an object is present, and zero elsewhere.

The primary intuition behind this is to spatially focus the model on the kind of object we would like to detect. While pure WSOD models tend to localize objects fairly well, they struggle estimating the objects' dimensions (Oquab et al., 2015). This is because the only limitation imposed on the model through the loss is the heat map sum constraint. Providing a few fully-supervised cases, however, encourages the model to concentrate more mass on just one prediction grid cell, instead of spreading it out to the neighbors. This not only reduces spurious, shallow detections of undesired objects, but also helps disentangling two instances next to each other that otherwise would likely be combined.

Furthermore, in cases where multiple kinds of objects appear in similar quantities, it helps reducing chances of the model detecting the wrong object class. This may particularly happen with presence/absence ground truths that naturally allow for high ambiguities.

4.3.3 Evaluating Positional Accuracies

Most fully-supervised object detectors predict bounding boxes, which can easily be evaluated through the intersection-over-union with the nearest ground truth box. In our case this does not apply, as we only predict heat maps. To assess the quality of the positional detection, we thus resort to a distance-based evaluation: we first threshold the heat map at a given value and subject all remaining locations to a Non-Maximum Suppression (NMS) stage to filter multiple hits of the same object. The assessment then follows the principle of Kellenberger et al. (2018c), which calculates the Euclidean distance of the respective

prediction cell coordinate to the nearest ground truth location, and only accepts it as a true positive if it is below a maximum distance and if the ground truth object has not been detected by another grid cell (in which case it counts as a false positive).

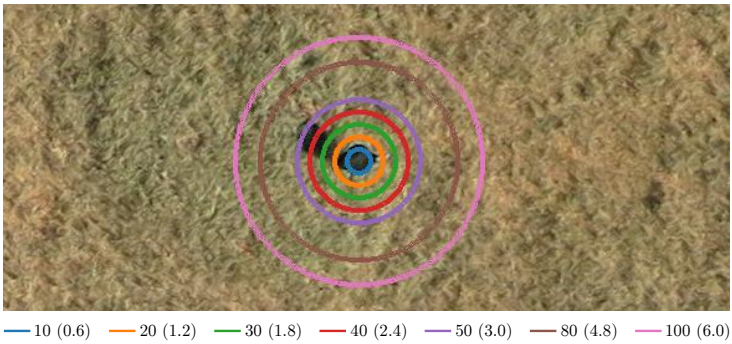


Figure 4.3: Concentric circles with varying radii over an animal in pixels. Equivalence in meters is reported in parentheses. Too small evaluation distances (blue) require models to precisely pinpoint the animal; too large (pink) risk including neighboring predictions. If we are also satisfied when the model predicts *e.g.* the animals' shadows, a radius in-between may be a viable choice.

This raises the question for the maximum distance threshold to be used. Especially in WSOD, we cannot expect the model to always perfectly predict the center position of an object. However, depending on the task, a pixel-precise localization may not be necessary after all, such as in our case of animal census. Consider Figure 4.3, where circles of different distances are shown. The smallest circle (blue) only encompasses just a part of the animal; this poses a severe restriction, as detectors are required to perfectly pinpoint the objects all the time. However, we would already be satisfied if the model detected *e.g.* the animal shadow, which lies outside the blue circle. Such a detection is thus only counted as a hit if we increase the distance threshold. Too large circles (brown, pink) in turn bear the risk of including false predictions outside of the animal. In essence, the positional tolerance is a function of the objective and the object size (in pixels). The effect of positional tolerance will be studied in the experiments.

4.4 Experiments

We now put the models with varying type and degree of supervision to the test. All in all, we compare models trained with the following levels of ground truth:

- Binary (baseline A): we regress the density map sum with the presence/absence of objects (Eq. (4.2)).
- Counts (baseline B): here, we provide the number of objects per image patch as a ground truth and use Eq. (4.1) as a loss.
- Binary + 1% (proposed A): this scenario adds full positional ground truth for 1% of the images (three images in our case) through Eq. (4.3), but presence/absence for the rest.
- Counts + 1% (proposed B): the same, but with counts as ground truth for the majority of images.
- Full dense (upper bound): we train the model with a complete set of positional ground truth (only Eq. (4.3) is used), and no counting or binary loss.

4.4.1 The Kuzikus Dataset

For evaluation we resort to a set of UAV images acquired over the Kuzikus game reserve in Namibia[1]. Kuzikus is a private-owned wildlife park that stretches across 103km^2 and is home to around 3000 individuals of multiple large mammal species, such as black rhino, ostriches, zebras, and various ungulates (Rey et al., 2017; Kellenberger et al., 2018c). In 2014, the SAVMAP consortium[2] imaged parts of the park with a SenseFly eBee[3] UAV, equipped with a consumer-grade camera at a flying altitude of around 100m above ground. This resulted in 654 nadir RGB images with a resolution of 4-8cm. An initial localization of animals was obtained through a crowd-sourcing campaign conducted by MicroMappers[4]; upon refinement of the labels, 1183 animals could be identified in the 654 images. This makes them a comparably rare sight and required manually examining every image for small targets. The dataset statistics are listed in Table 4.1; images are available at `https://doi.org/10.5281/zenodo.609023`.

[1] `http://kuzikus-namibia.de/xe_index.html`
[2] `https://lasig.epfl.ch/savmap`
[3] `https://www.sensefly.com`
[4] `https://micromappers.wordpress.com`

Table 4.1: Properties of the Kuzikus UAV dataset.

	# images			
Set	with animals	without	total	# animals
train	159	291	450	830
val	35	41	76	118
test	45	83	128	235

4.4.2 Model Setup and Training

Our model for all experiments is based on a ResNet-18 (He et al., 2016), pre-trained on ImageNet (Russakovsky et al., 2015), until the third-last layer: *i.e.*, we remove the original classification and average pooling layers and instead add two Multi-Layer Perceptrons (MLPs) that map from 512 to 1024 dimensions, and then to one (the heatmap), respectively. We replace the original BatchNorm layers with non-affine Instance Normalization variants (Ulyanov and Vedaldi, 2016), since we observed better stability for variable batch sizes during inference. Also, we reduce the first layer's stride to one to allow for a higher-resolution density map prediction. The density map output is scaled to $[0, 1]$ through a sigmoid activation.

To train the model we crop 16 random patches per epoch of size 512×512 from each UAV image in the training set and perform data augmentation by random flipping (both axes) and random 90° rotations. The model then predicts a density map of 32×32, which corresponds to a downsampling factor of 16. With our dataset's image resolution of around 4-8cm, one grid cell has a side length of roughly 0.6-1.3m, which still allows detecting animals below the average size of around 1.5m. This map gets summed and compared to the ground truth according to either loss function described above, depending on the degree of supervision (counts versus presence/absence of objects). We use mini-batches of four patches and employ the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-5} for the first 50, and 10^{-6} for the remaining 250 epochs. We also enable weight decay of 10^{-4}.

During inference we tile the images evenly into same-sized patches, evaluate them individually and stitch the predictions back together to one density map. For detection we filter the predicted heat map and retain locations with value 0.01 or greater. These are subjected to NMS with a search radius of 2 prediction grid cells (32 pixels; about 1.92m). Evaluation is then performed on a distance basis as in Kellenberger et al. (2018c), with thresholds of 30 pixels (ca. 1.8m) and 40 pixels (ca. 2.4m), respectively.

For the fully labeled images used by our proposed models we randomly select three images from the pool of images that contain at least one animal. The eventually selected images contain 1, 38, and 7 animals, respectively.

4.5 Results and Discussion

Table 4.2 lists Mean Absolute Error (MAE) and Mean Squared Error (MSE) values on the heat map sums over the test set. As can be seen, the count-based model with 1% full supervision yields the best score for both measures and is on a tie with the fully WSOD counts model in terms of MAE. Generally, it is not surprising that the count-based models outperform the others, even the model with full positional supervision, since they are the only ones explicitly trained on the objective of these measures, *i.e.* counting. The large MSE, but still reasonable MAE scores of the binary model imply that it predicts a few images wrong, but by a large difference (see Figure 4.4 top left). Since animals are a minority compared to background, this indicates that the model generally manages to localize them, but overpredicts them by also assigning high confidences to their spatial surroundings. This is not surprising, given that the binary model did not have any constraints other than to predict at least a mass sum of one whenever an animal is present. Adding just 1% of full supervision significantly improves the scores, although they do not reach those of the other models. In the case of the count-based models (Figure 4.4 top right), the WSOD-only model already performs similarly well as the fully-supervised one (Figure 4.4 bottom). However, here the 1% fully-supervised images have the effect of reducing false positives and hence improve localization performance, which can be seen in Figure 4.6.

Table 4.2: MAE and MSE values for the models on the sum of the density maps.

Model	MAE	MSE
Binary (baseline A)	1.96	22.29
Counts (baseline B)	**0.79**	1.63
Binary + 1% (proposed A)	1.22	4.99
Counts + 1% (proposed B)	**0.79**	**1.59**
Full dense	1.28	2.75

Figure 4.5 shows example images from the test set (top row), together with heat maps predicted by all five models. The images further contain the true animals' bounding boxes (cyan). As visible in the heat maps, the CNNs do indeed manage to predict precise locations of animals, even if trained with weak supervision. The binary model (baseline A) assigns particularly large masses to the animals, which corresponds to the observations made in Table 4.2 and Figure 4.4. Also, it is more prone to falsely detecting background objects like tree trunks, such as those in the left part of the left image. However, adding just 1% of positions (proposed A) significantly reduces both effects and dampens predictions around the animals, as well as around background locations. This confirms that a bit of strong supervision is enough to solve both issues (overprediction of animals and false detections of the mentioned background objects).

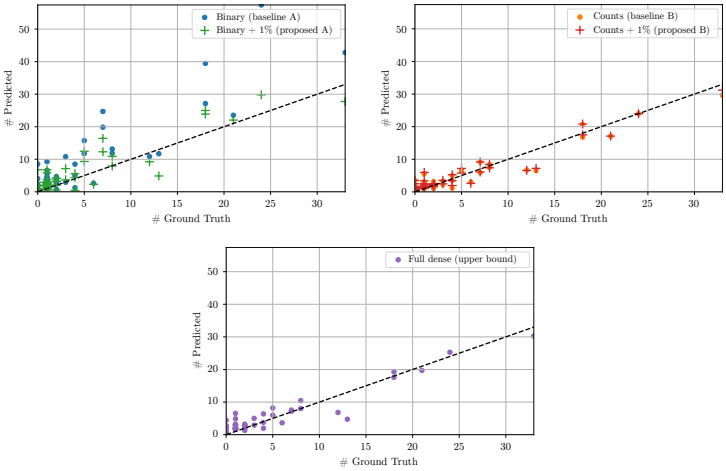


Figure 4.4: True animal counts versus predicted heat map sums for the two binary models (top left), the two count models (top right) and the fully-supervised model (bottom). In the binary case, 1% of full supervision (green) greatly reduces the overprediction of animal presence. For the count-based models the full supervision does not significantly improve the heat map sum, but instead raises the precision (see Figure 4.6). The fully-supervised model's heat map sum is similarly close to the optimum (dashed line).

In the case of the count-based models (baseline B, proposed B) the prediction strength is greatly reduced; activations over animals are smaller and with lower values. This shows that the counts pose more of a restriction to the model than the binary ground truth. However, despite this stronger learning signal also this baseline commits quite a few mistakes in the form of false positives, especially in the left image of Figure 4.5. Again, adding 1% of full supervision reduces the number of false detections. Visually, the upper bound (bottom row) nonetheless seems to provide the sharpest predictions (least spread in space around animals).

Figure 4.6 shows results for distance thresholds of 30 pixels (ca. 1.8m; left) and 40 pixels (ca. 2.4m; right), corresponding to the green and red circles in Figure 4.3, respectively. These distances are still fairly hard, as the models only get a small chance of predicting a true positive when *e.g.* concentrating mass on animal shadows. The curves show that, even under such conditions, all models manage to reach reasonably high recalls of ca. 70 to 90%, but with varying degrees of precision. In general, the binary model (baseline A) falls short the most, which goes in line with it not penalizing the wrong number of animals: despite NMS, it produces many false positives. The semi-supervised counterpart (proposed A) shows significant improvements in this respect, but does not come close to

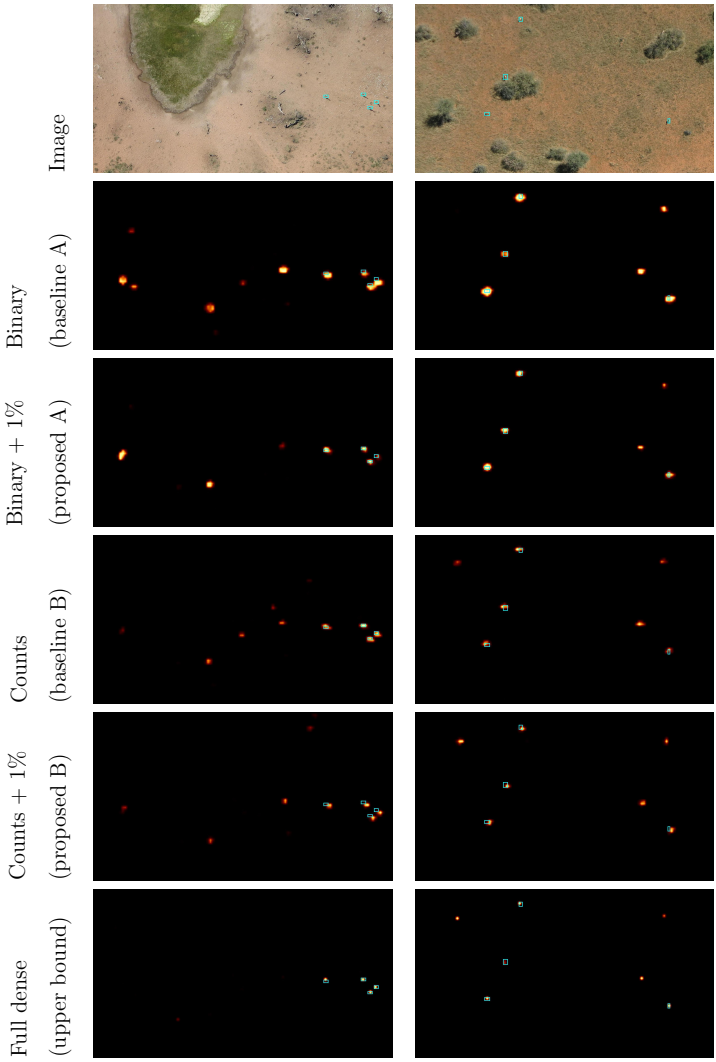


Figure 4.5: Crops of images from the Kuzikus dataset (top), along with predicted heat maps for all the models.

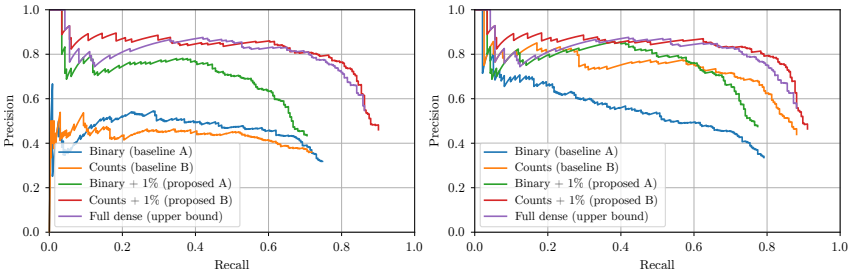


Figure 4.6: Precision-recall curves for the models on the Kuzikus test set, with a maximum ground truth distance of 30 pixels (1.8m; left) and 40 pixels (2.4m; right).

the levels of recall of the other models. Therefore, binary ground truth seems sufficient to get the models into a good direction, but not enough to really make them focus on the target objects. Higher percentages of full supervision might improve results in this respect, but we leave this to further studies.

For the count-based models, we already see a good performance with the WSOD-only model (baseline B), but only when using a more generous distance of 40 pixels. It seems that the counts-only model focuses more on shadows than on the animals, which corresponds to the slight spatial shifts of the heat map peaks observable in Figure 4.5. A threshold of 30 pixels is insufficient to account for such misplacements. However, we get a substantial improvement in the model with counts and 1% of positional ground truth (proposed B); in this case the model surprisingly is on par with the one trained on 100% positions (upper bound). This is even the case with 30 pixels tolerance, indicating that a few labels of strong supervision are enough to steer the model towards detecting the actual animal itself, rather than its shadow. It may seem surprising that this model actually outperforms the upper bound for recalls of around 80-90%. Our intuition on this phenomenon is that a full set of positions may actually be counterproductive, possibly due to limitations from the prediction grid: the heat map of the CNN is purposely downgraded in resolution, which may cause problems if *e.g.* the ground truth position lies on the border between two heat map grid cells. In those cases, assigning all mass to one of the two cells may be suboptimal for the model, especially if this happens frequently. Hence, the proposed model sees just enough positional ground truth to learn what to focus on, but not too much to cause problems due to spatial precision. All in all, the end-effect of this is that instead of having to densely label hundreds of images, weak image-based labels with a tiny fraction of full supervision gives equally good or even slightly better results, meaning that the labeling process becomes easier, faster, and thus less expensive, allowing for more or larger UAV datasets to be processed in a given time.

4.6 Conclusion

In this work we demonstrated how to reduce labeling efforts for training object detectors in Unmanned Aerial Vehicle (UAV) images. The proposed model resorts to Weakly-Supervised Object Detection (WSOD), which makes use of simple image-level labels like counts or presence/absence of objects. When augmented with just 1% of positional ground truth, the count-based variant effectively matches its counterpart trained on object positions in all images. Hence, we believe our strategy to be of major benefit to the labeling of large-scale UAV image datasets. To this end, further work is required on the role of the fully-supervised ground truth, in particular regarding the percentage of fully-supervised images, or else more sophisticated, perhaps model-driven strategies for selecting the images that require a positional ground truth. Also, model performance on objects of varying sizes could be worth investigating. Finally, improvements for the detection performance of the binary model could pave the way to lower the required labeling efforts even further.

Chapter 5

AIDE: Accelerating Image-Based Ecological Surveys with Artificial Intelligence

This chapter is based on:

> **Kellenberger, B.**, Tuia, D., and Morris, D. (in revision). AIDE: accelerating image-based ecological surveys with artificial intelligence. *Methods in Ecology and Evolution*.

Abstract

Ecological surveys increasingly rely on large-scale image datasets, typically terabytes of imagery for a single survey. The ability to collect this volume of data allows surveys of unprecedented scale, at the cost of expansive volumes of photo-interpretation labor. To this end, we present *Annotation Interface that Does Everything* (AIDE), an open-source web framework designed to alleviate the task of image annotation for ecological surveys. AIDE employs an easy-to-use and customizable labeling interface that supports multiple users, database storage, and scalability to the cloud and/or an arbitrary number of machines. Moreover, AIDE closely integrates users and a state-of-the-art Artificial Intelligence (AI) algorithm into a feedback loop, where user-provided annotations are employed to re-train the algorithm, and the latter is applied over unlabeled images to identify wildlife. These predictions are then presented to the users in optimized order. We conducted a user study on detecting large mammals in aerial images, in which we show that annotators were able to find almost four times as many animals in a given time if they were assisted by an AI algorithm. Furthermore, users were significantly more motivated in doing the annotation task, and significantly more confident in their own annotations, if guided by an algorithm. To the best of our knowledge, this is the first study that experimentally demonstrates a reduction in annotation time for ecological image surveys in a controlled setting. AIDE has the potential to greatly accelerate annotation tasks for a wide range of researchers employing image data. AIDE is open-source and can be downloaded for free at `https://github.com/microsoft/aerial_wildlife_detection`.

5.1 Introduction

Ecological research recently witnessed a tremendous increase in the usage of visual data. Motion-triggered camera traps produce hundreds of millions of images worldwide (Swanson et al., 2015; Weinstein, 2015), Unmanned Aerial Vehicles (UAVs) cover large areas with sub-decimeter resolution (Linchant et al., 2015; Baxter and Hamilton, 2018; Nowak et al., 2019), and in-field sound recorders capture spectrograms ("soundscapes") as a visual product in the terabytes for a single project (Servick, 2014). Such visual data enables non-invasive wildlife population estimation through spatial capture-recapture methods (Després-Einspenner et al., 2017) or aerial surveys (Hodgson et al., 2016; Rey et al., 2017; Kellenberger et al., 2018c), behavior analyses (de Kort et al., 2018), and habitat monitoring (Stark et al., 2018). However, this explosion of visual data comes at a cost: researchers spend weeks manually identifying species in images, or significant amounts of money to have the work outsourced. To this end, multiple image annotation interfaces have been proposed (Bubnicki et al., 2016; Krishnappa and Turner, 2014; Niedballa et al., 2016). While they facilitate data management, they typically lack *labeling assistance*, and still require humans to view every image to fully annotate a survey.

Recently, Computer Vision (CV) research has focused on automatically interpreting ecological imagery (Schneider et al., 2019; Willi et al., 2019; Tabak et al., 2019; Kellenberger et al., 2018c; Norouzzadeh et al., 2018), predominantly leveraging Convolutional Neural Networks (CNNs). However, employing these algorithms requires a high degree of expert knowledge, as well as a very large collection of labeled images for training. Moreover, CNNs are typically designed as self-contained, immutable entities: once they have been trained on a (manually annotated) part of the dataset, they remain static for the following phase, during which they are applied for prediction on the unlabeled portion of the data. This works if the dataset at hand shows similar properties for all images. However, in ecological applications, this is rarely the case, and datasets may contain differences large enough to make the algorithm fail. The consequence of this is that accuracy breaks down once it is applied to new, unseen data.

In this work we address both problems—the tedium of manual photo-interpretation and the constraints of Artificial Intelligence (AI) algorithms like CNNs—by unifying them into one labeling framework, which we denote *Annotation Interface that Does Everything* (AIDE). AIDE is a web-based, open-source collaboration platform that integrates an easy-to-use, versatile labeling tool and state-of-the-art AI algorithms into a feedback loop. Rather than training an AI algorithm offline, AIDE utilizes it to guide humans in the labeling process, and also make humans implicitly teach the algorithm what to look out for. We showcase AIDE in a study on animal detection in challenging UAV aerial imagery, where AIDE helps reduce annotation time by large amounts. The result is a platform that yields labels *and* re-usable AI algorithms simultaneously.

5.2 Methods

5.2.1 Overview

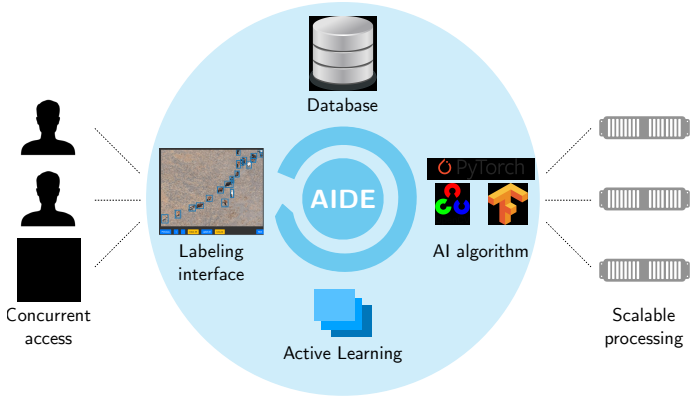


Figure 5.1: Overview of the workflow and components of AIDE.

AIDE is a web-based, collaborative annotation platform that includes humans and an AI algorithm in a loop, with both parties reinforcing each other for accelerated label retrieval. Figure 5.1 illustrates this loop and the key components of AIDE, including:

- *Labeling interface,* the primary access point for annotators and a window into the dataset to be annotated,
- *Database,* the storage solution for annotations and metadata,
- *AI algorithm,* which provides predictions on the images and learns from users through their annotations,
- *Active Learning (AL) criterion,* responsible for ordering the AI algorithm predictions in an optimal way.

The annotation process in AIDE iterates this loop until convergence, *i.e.* successful annotation of the entire dataset, or satisfactory prediction quality of the AI algorithm. The following sections outline this loop and the individual components.

5.2.2 Labeling interface

The labeling interface (Figure 5.2) is structured into two main parts: the main image viewer (left) and the list of label classes defined for the current project (right). The interface also contains a variety of controls for visualization (zoom, pan, a loupe, *etc.*) and interaction. Since the main target of AIDE is to obtain labels in the most efficient way, multi-step workflows, nested dialogues and pop-up messages have been avoided as much as possible.

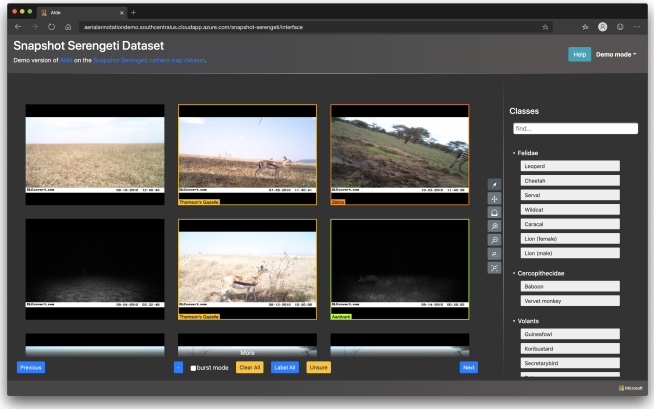


Figure 5.2: User interface of AIDE, with the main image viewer (left), annotation controls (below), and the list of label classes (right).

Annotation types

Image-based surveys vary significantly with their needs: for projects using camera trap data, image-wide labels denoting the presence or absence of a certain species might be sufficient, whereas a precise localization of animals is a key requirement for aerial imagery. To this end, AIDE supports a number of annotation types, namely image *labels*, *points* (with pixel coordinates), *bounding boxes*, and *segmentation maps* (where every pixel gets assigned a label). Figure 5.3 shows examples of the interface set up for the four currently supported annotation types. The annotation type for a project can be set by providing one single configuration parameter during setup, and AIDE will automatically adjust the user interface and tool set for it.

Annotating images

Users can create, modify, and delete annotations; the precise interaction paradigm depending on the annotation type. For instance, a click into an image either assigns it to a label (for whole-image labeling projects), or else sets a point at the specified position (for keypoint annotation projects). Clicking and dragging allows drawing and modifying bounding boxes, or painting or clearing a segmentation map.

Most of the labeling tools are assigned keyboard shortcuts, so that the user can keep their focus on the images, without having to look around to find the necessary tool. This also applies to the list of label classes, whose entries can be organized into hierarchical groups, collapsed, and searched. For instance, the search field can also be accessed through a keystroke – this way, users can keep the mouse cursor in the image view, and select the desired label class through simple keyboard operations, without having to scroll through the list of classes.

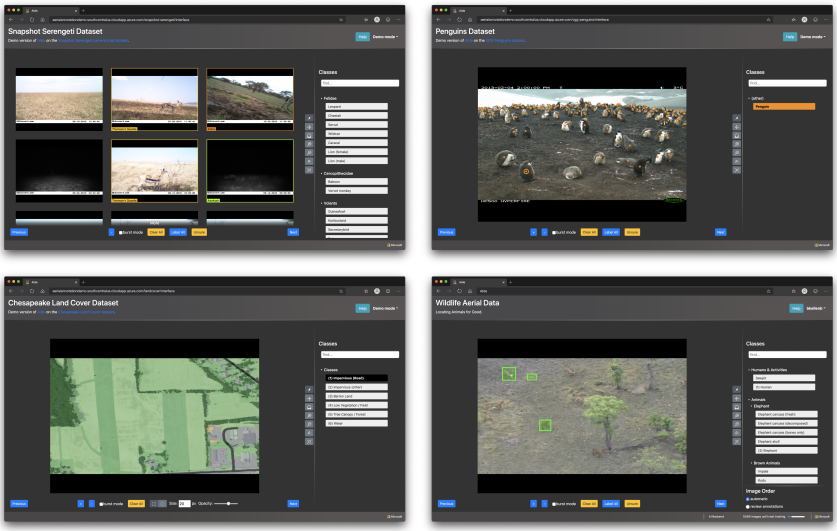


Figure 5.3: AIDE's labeling interface can be customized in many ways and supports multiple annotation types (clockwise, from top left): image labels, points, bounding boxes, and segmentation masks.

After a user annotates a set of images, clicking "Next" commits the annotations to the database (see Section 5.2.3 below) and presents a new set of images. Metadata related to the annotation process is stored as well, *e.g.* annotation author, image view count, date

and time of creation, time required, browser agent, window size, *etc*. Clicking "Previous" re-displays the image (or batch of images, depending on the configuration) the user has seen before and allows modifying annotations therein. Finally, the platform also supports re-visiting existing annotations, filterable by date and annotation presence/absence to skip empty images.

5.2.3 Database storage

AIDE stores annotations and metadata in a Relational Database (RDB), specifically *Postgres*[1], an open-source database system. RDBs enable concurrent (*i.e.* multi-user) access, scalability, and security on the one hand, but also facilitate tabular data download for further analyses on the other (Figure 5.4). Note that images are only referenced through the database, but stored as files on disk for easier organization.

	label class character varying	x real	y real	width real	height real	time created timestamp with time zone	time required [s] double precision	unsure boolean
1	Cattle	0.495668	0.49782	0.0803293	0.0968944	2019-08-06 08:15:55.459+00	19.703	false
2	Bird	0.494001	0.0910935	0.0339392	0.058074	2019-08-06 08:18:01.898+00	15.453	false
3	Zebra	0.506644	0.507558	0.0951205	0.104565	2019-08-06 08:20:46.701+00	15.065	false
4	Cattle	0.494375	0.502222	0.085	0.0844444	2019-08-06 08:24:52.877+00	13.358	true
5	Giraffe	0.514732	0.477192	0.130081	0.166241	2019-08-07 05:29:46.932+00	26.052	false
6	Buffalo	0.193129	0.0176897	0.0465653	0.0387569	2019-08-07 05:30:21.008+00	34.564	false
7	Buffalo	0.315766	0.757278	0.0782651	0.0658253	2019-08-07 05:30:21.008+00	46.771	false
8	Buffalo	0.0268138	0.70601	0.0440241	0.0873451	2019-08-07 05:30:21.008+00	49.671	false

Figure 5.4: AIDE stores annotations, references, and metadata in a Postgres database, which guarantees concurrent access, safety and security, and straight-forward tabular data download (screenshot from pgAdmin 4).

5.2.4 AI backend

At the heart of AIDE lies its built-in AI algorithm, which is intended to assist human annotators and accelerate the labeling process as a whole. Once users have annotated a sufficient number of images, AIDE uses the latest annotations to automatically re-train the AI algorithm in the background. This way, the algorithm can adapt to the dataset and improve prediction quality. In particular, the AI algorithm provides the following benefits:

1. Guidance: the algorithm can draw the labelers' attention to parts of an image that look like the objects of interest, which might otherwise have been neglected.
2. Assistance: in the database, user annotations and AI algorithm predictions are stored in different tables. However, the interface can be configured to automatically

[1] `https://www.postgresql.org`

convert algorithm predictions into annotations, which means that humans spend less time labeling targets that have already been identified by the algorithm (Figure 5.5). Every prediction that has been converted into an annotation can be modified, or deleted, by the human.

3. Acceleration: AIDE uses the algorithm predictions to determine the order in which images are shown to users (see Section 5.2.5).

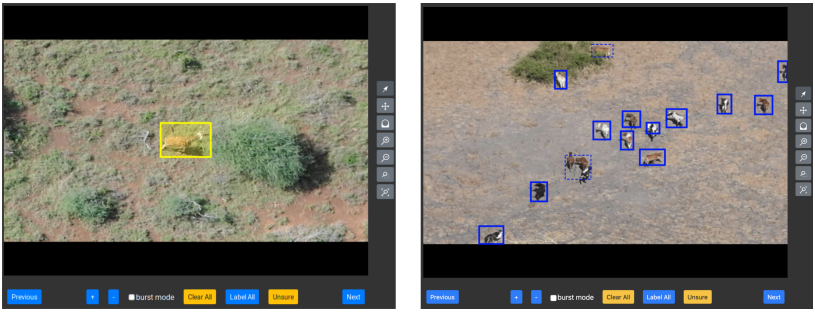


Figure 5.5: AIDE's AI algorithm can help reduce the annotator workload significantly. For example, the AI algorithm successfully detected the giraffe (left) and most of the cattle (right), which means that no interaction is required for the left image, and only a few missing animals need to be annotated in the right image.

The AI backend of AIDE has been designed to meet the expected needs of both annotators and project administrators:

Transparency—In principle, the AI algorithm is not directly visible to the user, apart from a small status panel (Figure 5.6). This means that annotators can focus on their task and are exposed to minimal distraction.

Non-intrusiveness—The AI algorithm supports annotators, but does not override them. Also, user *annotations* and algorithm *predictions* are strictly separated, down to the database. This guarantees that users always have the last word over the algorithm.

Ease of use—One major challenge of AI algorithms is that selecting the right algorithm for a new problem typically requires a substantial amount of machine learning expertise. AIDE attempts to overcome this problem by incorporating a variety of AI algorithms that can be accessed without modifying code, including:

- ResNet (He et al., 2016), a popular deep learning algorithm for image classification;
- The weakly-supervised point algorithm of (Kellenberger et al., 2019b), which has been specifically designed for animal detection from aerial imagery;
- RetinaNet (Lin et al., 2017), a popular deep-learning-based object detector.

In order to use one of these built-in algorithms, it suffices to provide their name and, optionally, a set of configuration parameters in the configuration file of AIDE, and everything else (algorithm training, saving algorithm states, prediction) will be handled automatically through the platform.

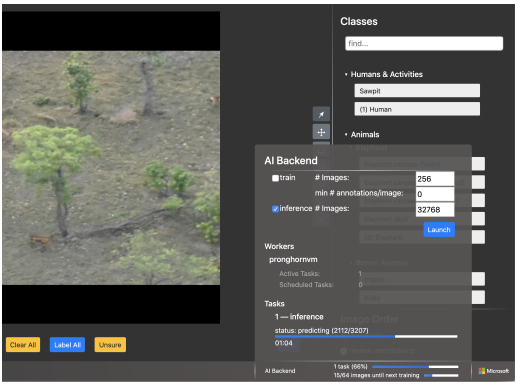


Figure 5.6: Status panel of the AI algorithm, as visible to the user. The manual controls for starting training and/or inference are only accessible to administrators.

Customizability—If the built-in algorithms are not suitable for a particular dataset, or if the user already has a custom algorithm available, AIDE supports integration of third-party algorithms. To do so, the user simply has to provide a few Python functions for training and evaluating and specify the path to the algorithm. The user is not limited to any algorithm library or even programming language, as long as the corresponding functions are implemented. Once configuration information is provided, AIDE will automatically embed the algorithm appropriately and provide all the data I/O (annotations and predictions to and from the database, images, algorithm states, *etc.*).

5.2.5 Active Learning for human-machine collaboration

In most AI workflows, an AI algorithm is trained once on parts of a dataset and then kept static during a prediction phase on the rest of the images. While this may work if sufficient data has been labeled, it is less than optimal for situations where the AI is expected to evolve throughout the labeling process.

Instead, AIDE employs AI in a so-called Active Learning (AL) loop (Settles, 2012; Kellenberger et al., 2019a): humans begin labeling images, and after a number of annotations have been made, these are automatically used to re-train the algorithm. The updated algorithm is immediately employed to predict images that have not yet been reviewed

by the annotators. All of this happens in the background and does not interrupt the labeling workflow. These newly predicted images are then directly considered through an AL criterion: for example, AIDE can be configured to prioritize images shown to the users that contain a lot of predictions, or predictions made by the algorithm with a very high confidence score. In the end, this means that more relevant images are shown to the user first, which accelerates the number of labels acquired in a given time, and keeps user motivation high.

5.3 User study

In this section, we evaluate the benefit of an AI algorithm through a user study employing the platform. More specifically, we assess the degree to which AL can accelerate the task of locating animals in aerial images collected from the African savanna. This is a highly challenging task: the areas covered by aircraft are generally vast, and animals are sparse, making the localization task a difficult needle-in-the-haystack problem (Kellenberger et al., 2018c).

5.3.1 Study area and data

We use a dataset of UAV images that have been acquired over the privately-owned Kuzikus wildlife reserve[2]. Kuzikus is home to approximately 3000 large mammals, including black rhino (*Diceros bicornis*), Burchell's zebra (*Equus quagga burchellii*), and various wildebeest species. Two surveys were conducted by the SAVMAP consortium[3] in 2014 and 2015, resulting in 654 (2014) and 3254 (2015) images of $\geq 4000 \times 3000$ pixels. In an initial step, both datasets were manually labeled by using conventional tools, resulting in 1183 (2014) and 646 (2015) animal bounding boxes (see Kellenberger et al. (2019a) for more details). We then split the images into patches of size 800×600 around animals for the 2014 data, and on a regular grid with 25% overlap in the x and y directions for the 2015 images. Doing so alleviated the users participating in the study from having to zoom in and out. This resulted in 11,935 patches for the year 2014, and 326,335 for 2015.

[2] `https://kuzikus-namibia.de/xe_index.html`

[3] `https://www.epfl.ch/labs/lasig/research/projects/savmap`

5.3.2 Experimental setup

15 volunteers participated in the user study. They were instructed to draw bounding boxes around animals (without species identification) in a given time span. We set up two instances of AIDE as follows:

- *Without algorithm:* the first instance contained all patches of 2015, without any existing annotations, predictions, or AI algorithm. The patches were shown to the user in completely random order.
- *With algorithm:* the second instance featured exactly the same dataset, but this time also contained predictions made for each patch by an AI algorithm. Specifically, the algorithm was a RetinaNet that had been trained on the 2014 portion of the data. In this setup, the image patches were ordered according to the *confidence* of the algorithm predictions made in the image (*i.e.*, if the algorithm predicted one or more animals in a patch with a very high confidence value, this patch would be shown with higher priority).

The user study workflow then proceeded as follows:

1. Users could get familiarized with the controls of AIDE on a demo instance (*10 minutes*).
2. In a first round, eight users were randomly assigned to the first instance (without algorithm), seven to the second, and all were instructed to draw tight bounding boxes around animals (*15 minutes*).
3. After a two-minute break, the users switched to the other instance and repeated the same task (*2 + 15 minutes*).
4. Finally, the annotators filled in a survey to provide feedback on the two modes (*5 minutes*).

5.3.3 Results and discussion

Figure 5.7 shows the number of animals correctly identified by the annotators. A bounding box was deemed as "correct" if its intersection-over-union with the closest ground truth box was greater than or equal to 0.5. Since only a fraction of all images were shown in the time spans, the total number of animals in each scenario is dictated by the image order (*i.e.*, it is higher if more images with animals were shown first). This upper bound is drawn with dashed lines for each scenario.

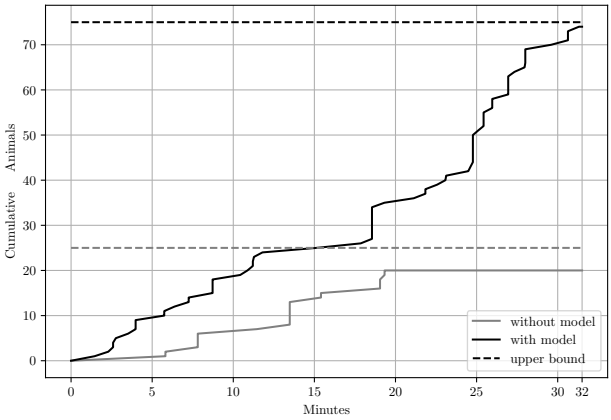


Figure 5.7: Number of animals found by the annotators over time for both setups (solid), as well as the actual number of animals in the images (dashed).

The results show that the annotators found 3.7 times more animals if they were assisted by an AI algorithm (74 vs. 20). Essentially, the AI algorithm helped prioritize the images that were more likely to contain animals. Also, by providing predicted bounding boxes, the algorithm further guided annotators toward objects that were more likely to look like animals.

Figure 5.8 shows the number of images viewed for each setup over the study session. At first glance, the setup without an algorithm (gray) provided a higher throughput than the algorithm-assisted instance. However, this can again be traced back to most of the images containing no animal: inevitably, annotators have to invest time in drawing boxes, but can simply click "Next" if the image happens not to contain any animals. Hence, the higher number of images viewed comes with the cost of the images being of lower importance. In the worst case, this could mean that annotators would have to review *all* images to find all of the animals, as some might only appear at the end.

On the other hand, CNN-based algorithms have been shown to be able to find about 80% of the animals in these datasets (Kellenberger et al., 2019a). In the present case, the algorithm predicted animals in 14,751 patches (4.5%). In essence, this means that in this case, users would only have to annotate 4.5% of the images to find most of the animals. This number can be reduced further with *e.g.* a custom AL criterion (Kellenberger et al., 2019a).

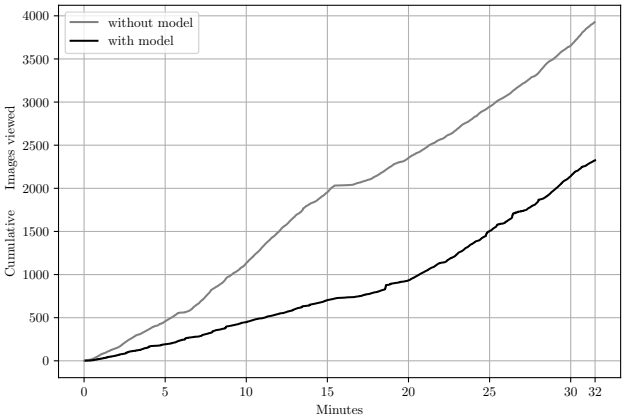


Figure 5.8: Number of images viewed by the annotators over time.

In a last step, we asked the volunteers to answer a few questions on the annotation experience in using the two instances (with and without algorithm). Figure 5.9 shows the annotator responses to two of the survey questions. The results indicate that annotators found it significantly easier to localize animals if assisted by an AI algorithm, and were significantly more confident in their own annotations. The remaining two questions and results are shown in Figure 5.10 in the Appendix.

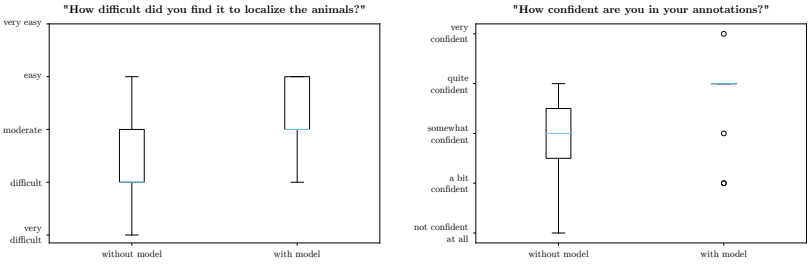


Figure 5.9: Survey responses for two of the questions for each of the setups.

5.4 Conclusion

Ecological research increasingly relies on large-scale visual datasets, which can dramatically scale the spatial coverage of wildlife surveys, but require tedious and expensive photo-interpretation of the images acquired. Artificial Intelligence (AI) algorithms, in particular Convolutional Neural Networks (CNNs), have demonstrated high potential for accelerating this manual work. However, the need for Computer Vision (CV) expertise and large training sets so far prevented broad adoption of these algorithms.

In this study we presented *Annotation Interface that Does Everything* (AIDE), an open-source web framework that integrates a flexible and easy-to-use annotation platform with a state-of-the-art AI algorithm. AIDE is a versatile labeling tool that offers a high degree of customizability, support for various annotation types, and support for multiple users. It is also one of the first annotation platforms that employs an AI algorithm to assist annotators in their task. Critically, AIDE employs AI in a way known as Active Learning (AL), with humans and the machine working hand-in-hand: humans provide annotations the algorithm can learn from, and the algorithm returns suggested predictions and prioritizes images with respect to their relevance. In a user study on the task of localizing infrequently occurring animals in aerial images, we have shown that when an AI algorithm is employed, annotators find almost four times as many animals in the same time compared to a scenario without an algorithm. Additionally, annotators were significantly more confident in their annotations, and more motivated in the labeling task.

AIDE is an open source-platform and free to use. The source code is available at `https://github.com/microsoft/aerial_wildlife_detection`.

Acknowledgments

The authors would like to acknowledge the SAVMAP consortium (in particular Dr. Friedrich Reinhard of Kuzikus Wildlife Reserve, Namibia) and the QCRI and Micromappers (in particular Dr. Ferda Ofli and Ji Kim Lucas) for the support in the collection of ground truth data for the user studies. We would also like to thank Dr. Howard Frederick for providing the image data used in parts of the screenshots shown in the chapter. Finally, we gratefully acknowledge the support of the NVIDIA Corporation with the donation of a Titan V GPU used for this research.

Appendix

5.4.1 Survey Questionnaire

The questionnaire in the user study (Section 5.3) comprised the following two additional questions and yielded results as shown in Figure 5.10.

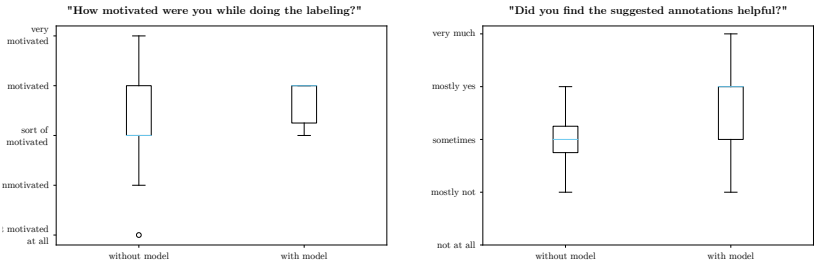


Figure 5.10: Additional questions and results asked in the user study.

Chapter 6

Synthesis

6.1 Main results

This thesis attempts to advance wildlife conservation using machine vision, with focus on monitoring animals through aerial imagery. The aerial perspective offers promising improvements over conventional census modes like foot surveys. However, the vast amount of imagery produced by *e.g.* Unmanned Aerial Vehicles (UAVs) cannot economically be processed through purely manual photo-interpretation. Machines employing Computer Vision (CV) methods may provide an answer to this challenge: they can process large amounts of data, nowadays even while performing complex tasks like animal detection. However, they struggle with the small size and scarcity of animals in aerial images; they cannot easily generalize across different datasets, let alone wildlife reserves; and they still require a reasonable number of manually drawn inputs to training models. This thesis provides strategies to facilitate, respectively overcome all of those problems, and shows through case studies that all challenges see significant improvements if addressed with appropriate methodologies. In this context, a recurrent finding was the importance of *interactivity*, which combines the strengths of both the machine and human operator(s) in an optimal way. The end result is an interactive annotation framework that incorporates the major findings of this thesis and already sees application in multiple cases at the time of writing of this manuscript. The main challenges of interactive animal census monitoring were described in four research questions in Chapter 1, and will be answered below.

6.1.1 How can state-of-the-art Deep Learning (DL) models be trained for the task of aerial image-based animal localization?

A model capable of accurate animal localization forms the core of any machine-assisted census environment. Accuracy in this context refers to simultaneous high recall (*i.e.*, high percentage of animals successfully identified) and high precision (*i.e.*, the percentage of detections being true positives). Determining precision and recall in turn requires establishing criteria at which a detection can be counted as a true positive. To this end, Chapter 2 introduced a census-oriented evaluation protocol, which validates model predictions with priority on abundance, rather than absolute spatial correctness. This was further adapted for the weakly-supervised setting in Chapter 4. Also, in preparation for interactive scenarios like in Chapters 3 and 5, tile-, or image patch-based evaluation criteria were examined that focus on the required annotation input, rather than model performance.

With established evaluation criteria, the primarily observed problem with both conventional CV object detectors and off-the-shelf Convolutional Neural Networks (CNNs) was a severe drop in precision at high recalls. In other words, models would vastly overestimate the presence of animals, thereby rendering predictions useless. Chapter 2 proposed a se-

ries of training recommendations, including artificial class balancing, gradually increased complexity exposure through curriculum learning, and attenuation of spurious predictions through hard negative mining and a dedicated border class. This resulted in a CNN-based detector that was able to predict animals with a precision of initially 30% at 80% recall, compared to less than 10% of the previous state-of-the-art (three-fold improvement). Further improvements, such as the employment of instance normalization, were implemented throughout Chapters 3 and 4, which eventually led to a precision of 80% at 80% recall (eight-fold improvement). In other words, this means that CV models are now able to localize 80% of the animals in UAV imagery, and that only two out of ten predictions are false positives. Although the predominantly used model is a non-standard adaptation of ResNet, originally a CNN architecture for image classification, it was found that more conventional state-of-the-art object detectors like RetinaNet employed in Chapter 5 benefit in similar ways and show equally significant improvements and performance if trained with such recommendations. This shows the importance of carefully selecting appropriate training routines, rather than just adding more data to train a bigger model.

6.1.2 What are the necessities and possibilities of CV in the context of animal population monitoring through census repetitions?

Long-term population monitoring involves recurrent census estimations, and therefore vast amounts of data generation and interpretation on a periodical basis. This inevitably multiplies the problems of data handling. In addition, it complicates reusing detection models due to differences in characteristics of datasets, a phenomenon known as domain shift. Bridging the gap from one census campaign to the next requires models that are able to generalize across domains. Of central interest in this context is the *Domain Adaptation* (DA) criterion presented in Chapter 3, denoted as *Transfer Sampling* (TS). Conventional DA strategies in CV were found not to be operative in a UAV-based census setting where animals are scarce. TS in turn was specifically designed for problems with large class imbalance like aerial censuses. To this end, the criterion employs Optimal Transport and a sampling strategy that leverages the detector model's superior performance in the source domain. Both ingredients combined effectively make the adaptation process immune to class imbalance. In addition, TS directly incorporates annotators through Active Learning (AL), thereby significantly reducing the risk of false class assignments. Effectively, TS has the promise for highly efficient long-term monitoring: once an initial detection CNN like the one presented in Chapter 2 has been established, it can be reused and adapted with a minimal amount of supervision for any following image campaign.

6.1.3 In which ways can the workload of human annotators in censuses and census repetitions be reduced, respectively optimized?

With high-resolution data available from UAVs, the most time- and labor-intensive aspect of wildlife censuses can be attributed to photo-interpretation and image annotation. The detector CNN presented in Chapter 2 provides an initial reduction in the amount of required photo-interpretation. Annotations were treated as training labels for the model, and the model in turn could be employed to predict animals in unseen images. Although the primary focus of this chapter was not on minimizing the number of labels required, the proposed detection model basically enabled a reduction of the annotations to 60% of the original dataset.

Chapter 5 extended this machine vision assistance by integrating it with humans in an AL loop right from the start of the annotation phase. AL directly intends to reduce the required amount of training data to an optimal subset. This chapter exemplifies the AL paradigm and is one of the first to evaluate its usefulness in the context of aerial wildlife censuses. Experiments showed that employing machine vision assistance in the labeling process resulted in a four-fold reduction of the required number of images to be labeled by annotators.

A third aspect of simplifying user intervention addresses census repetitions. To this end, Chapter 3 found that adapting trained models for new censuses results in even stronger workload reductions. The TS criterion mentioned above plays one part of this role, as it was found to prioritize more animal-alike—and therefore more relevant—predictions for DA. The other part is down to the window cropping strategy, which selects an optimal patch around the sample selected by TS. This augments the number of available labels from the target set, and also is more intuitive for annotators. Although the workflow requires label inputs from the user(s), experiments in Chapter 3 have shown that less than half a percent of images need to be annotated for TS and window cropping to find 80% of the animals in census repetition scenarios. In other words, the model solves a "needle in the haystack" problem by prioritizing only the most relevant samples, and the vast majority of images that does not contain animals does not even need to be presented to the annotator.

Finally, the idea of optimizing human workload was further studied in Chapter 4. Rather than just reducing the absolute number of annotations, this chapter focused on simplifying them in such a way that overall less complex user input is required, and more images can be reviewed and annotated in a given amount of time. The proposed path was to employ Weakly-Supervised Object Detection (WSOD), which trains object detectors by only telling them *if* a certain training image contains animals, rather than *where*. Experiments showed that the model from Chapter 2 was indeed able to localize the animals to a fair degree with exclusively image-wide presence/absence ground truth. However, if

just one percent of full spatial supervision was added, the prediction accuracy was on par with the variant trained on entirely spatial supervision. One percent corresponded to three fully annotated images in the training set, which is a virtually negligible annotation overhead, especially if the other 99% of the training images only need to be labeled with simple presence/absence of animals. In sum, the methodologies presented throughout all chapters resulted in a reduction of annotation efforts to a small fraction of the dataset, and in a strong simplification of the annotation task itself, both for instantaneous animal detections and census monitoring settings.

6.1.4 How can machine vision strategies and methodologies for animal censuses be conveyed to the target audience in an applicable manner?

Machine-assisted wildlife conservation is a multidisciplinary field: censuses may be requested or initiated by conservation ecologists, governments, or other organizations; the aspect of machine assistance involves CV, which in turn relies on methods from computer science and Machine Learning (ML); and beyond conservation, herd location and abundance data might further be of interest for behavioral ecology. Ultimately, however, the tools and findings throughout this thesis most directly target humans in the role of annotators. Although various strategies were proposed that significantly reduce their required labeling efforts (Chapters 2 to 4), it nonetheless continues to be a major aspect of censuses work- and time-wise. As such, it is imperative to not only propose, but also *deploy* these strategies to assist annotators. However, this needs to be done in such a way that *technicalities are hidden* from the annotators: machine vision assistance should be there to assist humans, not to interfere with, let alone override them. Also, in an applied setting, there should be little to no requirement for end-users to devise their own code, *e.g.* for data im- and export, CV model training and optimization, and the like. All of this would greatly distract from the annotation process, and may result in users having to figure out ways to get CV models to work on their dataset, a task that requires an uneconomical amount of time as well as expert knowledge in computer science. However, at the same time, some conservationists may do indeed have access to CV models and/or knowledge to train them, but would like to include them into the annotation process, *e.g.* through AL.

The end-effect of this is that making machine vision technology available to end-users requires managing a balancing act between two paradoxical targets: hiding complex implementation details from the annotation process on the one hand, but also enabling sufficient flexibility for users who wish to do more on the other hand. With these targets in mind, Chapter 5 presented *Annotation Interface that Does Everything* (AIDE), which is one of the first platforms that employs machine vision assistance in the labeling process right from the start. To the end-users, AIDE exposes a versatile but easy-to-use web interface that allows annotators to focus on their task. But in the background, it

includes a complete suite for training and employing state-of-the-art CV models on the dataset and task (object detection, image classification, *etc.*) at hand. AIDE has some of the most common DL-based models built-in, as well as the WSOD model presented in Chapter 4, and is able to handle data management and model training all by itself. If required, users are able to configure the built-in models to their needs, or else include their own algorithms into the platform. Then, rather than just training models in the background, AIDE tightly integrates them into the labeling process: user annotations serve for training, and resulting updated models are employed to predict and prioritize unseen data through a customizable AL criterion. This workflow prioritizes the most relevant images to be labeled by the annotators, and with that significantly accelerates the entire annotation process (*cf.* Chapter 3). The final product is a versatile software suite that simultaneously offers high ease of use as well as high flexibility. At the time of writing, AIDE is actively used within the Tanzania Conservation Resource Centre[1] and has further been presented to and considered by the Tanzania Wildlife Research Institute[2], WildMe[3], Conservation Metrics, Inc.[4], Hensoldt GEW[5], Vulcan[6] as well as other, non-enterprise entities and organizations. AIDE is an actively developed, open source project and available under `https://github.com/microsoft/aerial_wildlife_detection`.

6.2 Reflection and outlook

Species and habitat loss of vertebrates are at an all-time high since environment alterations through humans, and projections reveal a likely acceleration of this trend in the current century (Powers and Jetz, 2019). Therefore, measures to countersteer this trend are in high demand. In this context, it is important that these measures are applied systematically, which requires assessments of current species abundances. This thesis proposes to conduct censuses using technology, including UAVs for inexpensive and safe data acquisition, and machine vision for efficient animal localization and abundance retrieval. Experimental results of machine vision technologies proposed throughout the chapters of this thesis are highly promising. However, most of the work presented still requires follow-up investigations and improvements. Furthermore, the machine vision context offers a number of opportunities that are potentially useful for censuses, but have not been addressed in this thesis. In addition, technology-assisted censuses have influences and impacts beyond machine vision, which implies that better solutions could be found if other disciplines, such as UAV technology and ecology, are more directly considered and

[1] http://www.tzcrc.org
[2] http://tawiri.or.tz
[3] https://www.wildme.org
[4] https://conservationmetrics.com
[5] http://www.gew.co.za/hensoldt-south-africa
[6] https://www.vulcan.com

integrated. Overall, further work needs to be done specifically for machine vision, and also in a broader context across disciplines. The remainder of this chapter attempts to provide an overview over both scopes accordingly.

6.2.1 Further work in the machine vision realm

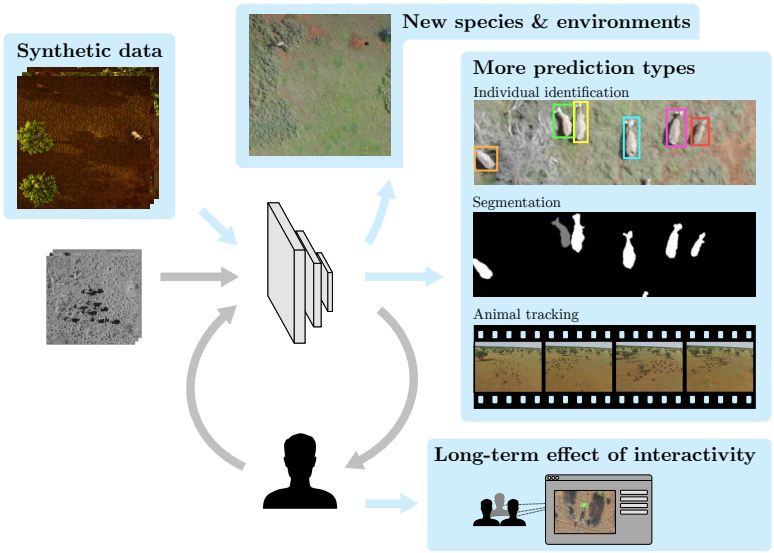


Figure 6.1: The machine vision context offers potential for a number of follow-up studies that could result in better conservation efforts. A few of them are highlighted here in blue.

Figure 6.1 shows the interactive annotation framework presented in this thesis (gray), and a subset of potential research fields to follow up on in terms of machine vision (blue). The two main reoccurring patterns throughout this thesis, namely ML/CV models and interactive systems, could profit from further works in a number of ways, including:

- Applying models for *new species and environments*: most of the experiments in this thesis were limited to large mammal detection in the African savanna, but the methodology might be applicable—and is most certainly required—in other ecosystems, and for previously unaddressed species.
- Providing *more prediction types*: in addition to mere animal localization, new model outputs could greatly improve the understanding of the environment (see also Section 6.2.2 below).

- Exploring new data modes like *synthetic data*: doing so not only has the potential to improve model performances, but might also enable new target applications for which model training on real-world data is difficult or impossible.
- Assessing the *long-term effect of interactivity*: Chapter 5 presented one way of interactively performing censuses and has shown the potential of this setting in a user study. The eventual usefulness of such systems in the long run still needs to be investigated.

New species & environments

The methodologies and workflows proposed in this thesis have primarily been tailored towards large mammal detection in aerial images. While the details of model training and definition depend on the species in some sense (*e.g.* on the animal abundance and densities; *cf*. Chapter 2), the methods have potential to be applied to other targets and environments as well.

For example, Figure 6.2 shows an application of the weakly-supervised model from Chapter 4 on the detection of birds in Western Africa (manuscript in preparation). Compared to large mammal censuses with scarce animals, bird colonies pose the opposite problem, in that individuals are extremely abundant. In both cases the end-effect for humans is a prohibitively expensive amount of manual labor, unless assisted by machine vision. In the birds case, this is made possible through an extension of the WSOD model to work with multiple species, as well as polygonal ground truth, instead of simple per-image presence/absence information. The result is that despite the fundamental differences in class distributions between the birds and mammals of this thesis, it is possible to detect individual birds to a high degree of precision by reducing the amount of required ground truth to an absolute minimum.

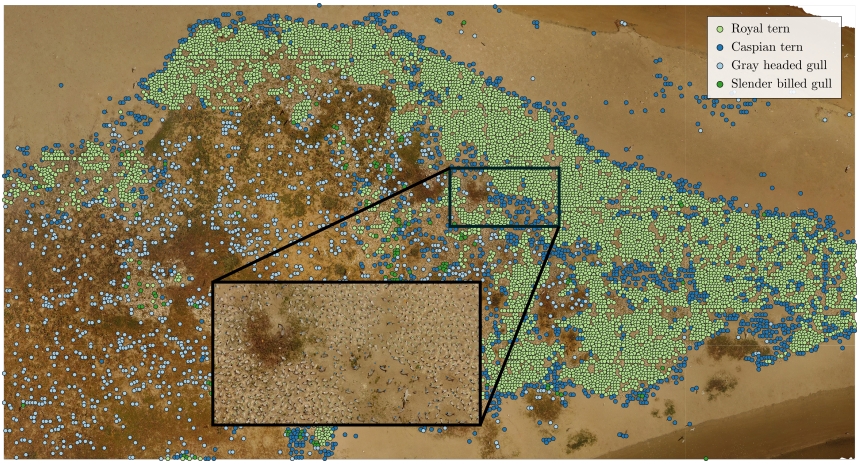


Figure 6.2: Example application of the WSOD model presented in Chapter 4 on the detection of multiple bird species. In this case, the orthomosaics contain more than 70,000 birds, but, thanks to weak supervision, the model only needed 20 point annotations per class and a few polygons for training. Together, these ground truths required less than 30 minutes of annotation time.

Similarly, further investigations could be conducted in the model adaptation context. Beyond the outlined setting of census repetitions within a wildlife region (Chapter 3), having models be able to generalize *across regions* could potentially enhance detections in a collaborative way. For example, an extension of the TS criterion presented in Chapter 3 for multi-species support might provide even better adaptation results, perhaps with the species acting as an ensemble and providing better separability between animals in general and falsely detected background objects.

On a different note, users of census systems might not always know at the start *which* kind of species they will encounter. This again also applies to adaptation of models to new regions. For example, the African elephant is a species that does not occur in the Kuzikus reserve, but is present in other wildlife areas. Adapting detection models from Kuzikus to such places would require the option to add new species to a pre-trained model. At this point, this is not possible for the models presented in this thesis. However, this mode has been studied in CV, and fall into the categories of *few-shot learning* (Snell et al., 2017) and *lifelong learning* (Srivastava et al., 2019).

An alternative approach to adapting models on-the-fly has been proposed by Beery et al. (2018) for camera trap imagery: rather than training individual animal detectors per ecosystem or wildlife park, the authors employ one general, species-agnostic detection

model that can be applied regardless of the image origin, followed by smaller classifiers that are tuned for the region and ecosystem at hand. The result is a high-quality animal detector that can be applied with low adaptation efforts to a large variety of ecosystems globally. It is unclear whether this strategy could be applied to animal localization in aerial imagery, but if possible, it could open new avenues for upscaling censuses to significantly larger and more diverse extents.

More prediction types

Besides detection and species classification, CV offers more and also more advanced prediction types that could be applied with benefits to aerial animal conservation. A first example is individual identification, where not only the species, but separate *members* of an animal population are identified. Unlike animal species classification, individual detection requires the capability of models to extract feature descriptors that highlight the subtle, unique differences between individuals, rather than just species or animals in general. Individual identification has been extensively studied for humans in the form of person re-identification (*e.g.* Zheng et al. (2015)), and to a lesser extent for certain animal species. For example, Lahiri et al. (2011) identify individual zebras based on differences in the striping patterns of their fur and skin. Crall et al. (2013) extend this idea to other species, such as giraffes and leopards. These works generally rely on distinctive patterns on the animal torsos, and therefore require high-quality imagery taken from a close-up perspective near the animal. A viable image source that has been used for close-up views are photographs taken by tourists or ground forces (Crall et al., 2013; Foglio et al., 2019). With such imagery, individual identification could be fruitfully extended to other species that exhibit more subtle body characteristics, such as wrinkles around eyes (Patton and Campbell, 2011) in the case of rhinos.

A second type of prediction that could be explored for conservation efforts is known in CV as *segmentation*. This includes very high-resolution predictions, typically in the form of one label per pixel. In some sense, it can be thought of as *e.g.* the detection model presented in Chapter 2, but rather than downsampling and predicting a label for a group of *e.g.* 16×16 pixels, a model would attempt to estimate for every pixel individually whether it shows parts of an animal or not. This mode is known as *semantic* segmentation. Another example is *instance* segmentation, where pixels over animals are assigned individual labels, depending on the animal instance they belong to. This might be necessary when animals are standing closely together. Segmentation outputs could provide detailed insights into the size, shape and posture of animals, and with that are potential indicators for an animal's health. Segmentation has been investigated for cattle farming (Qiao et al., 2019), and could likely also be applied for endangered wildlife individuals, in order to get proxies on their physical health.

A third CV output addresses animal trajectory estimation, also known as *tracking*, where the position (and extents) of an animal is recorded throughout a series of images, or a video. Unlike detection in still images, tracking requires repeated estimations of the position and extent in a temporally consistent way. This involves a new set of challenges that are not of primary relevance in the detection context. For example, moving animals might change orientations and pose throughout time, or they could temporary be occluded if they *e.g.* move underneath a tree. Object tracking itself has been extensively studied in CV (Wu et al., 2015; Kristan et al., 2015, 2018), and also the reoccurring problem of small objects has been addressed with tracking in mind (Risse et al., 2017; Haalck et al., 2020). The context of wildlife areas observed from an aerial viewpoint offers a number of application opportunities for tracking, maybe even in combination with individual identification for animal behavior studies, or other secondary outputs (see Section 6.2.2 below for examples).

Using synthetic data for model training

A slightly less objective-dependent CV topic addresses the training of models using generated, or synthetic data. In CV, synthetic data typically refers to images of artificially created landscapes, rendered using computer graphics techniques like ray tracers. The intent to employ synthetic data for model training originates from CV applications where natural data or labels cannot be acquired. A common example comes from the field of self-driving cars, which have to be trained to detect and avoid accidents. The creation of training images and labels for such dangerous situations clearly cannot be enforced, but can be simulated in virtually infinite amounts using artificial landscapes and rendering engines (Ros et al., 2016). The two main bottlenecks in employing synthetic data for model training are the involved process of creating artificial landscapes in the first place, and lack of realism due to limitations of the rendering engine's capabilities. However, this second problem has been shown to be addressable to some extent using DA techniques (Hoffman et al. (2018); *cf*. Chapter 3).

In the realm of aerial wildlife conservation, works employing synthetically created data are virtually inexistent, with the exception of the *AirSim-W* environment presented by Bondi et al. (2018a). An example rendering of AirSim-W from an overhead perspective is shown in Figure 6.3. Synthetic data could be useful for wildlife censuses in multiple ways. An example where ground truth is too dangerous or prohibitive to be collected is poaching prevention: UAVs are not only used to detect animals, but also to spot and interfere with poachers (Bondi et al., 2018b). For the physical protection of animals and workforce, poaching events inevitably have to be simulated. In addition, synthetic data could also be employed to reduce the number of annotated training images—a model pre-trained on rendered images might need less fine-tuning for the target dataset at hand.

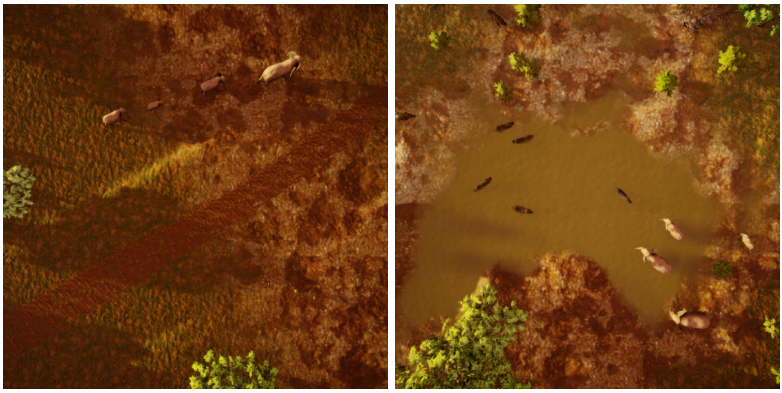


Figure 6.3: Synthetically rendered imagery of wildlife areas, such as the *AirSim-W* environment shown here (Bondi et al., 2018a), might help to fill the lack of available, large-scale UAV datasets for animal censuses. Image courtesy of Derek van de Ven.

Long-term effects of interactive censuses

A final machine vision-related aspect to be deepened further addresses interactive annotation platforms, such as AIDE presented in Chapter 5. Experiments have shown that including annotators and a model in an AL loop from the start results in a significantly higher throughput and less required annotation time for labeling a dataset. However, it cannot be ignored that the labeling sessions in the experiments lasted for a comparably short time. Although extrapolations to longer labeling sessions from these investigations may be realistic to some degree, they are likely not sufficient for determining the *long-term* usefulness of interactive annotation platforms. It could well be that AL only helps in the beginning of a labeling session, but not afterwards.

On another note, further tests need to be conducted on the *user satisfaction* in labeling through an interactive platform. The experiments in Chapter 5 included a small survey to this end, and showed that users were significantly more motivated when they were assisted with model predictions, likely due to the lower degree of monotony in the images. However, as before, these results might not always hold for long-term annotation sessions—for example, repeatedly spurious model predictions could at some point become a nuisance for humans, or else the model could mislead them by predicting animals over background objects that look too similar to animals. Hence, further work is needed that tests interactive platforms like AIDE in the long run, for example through a case study over a large wildlife reserve and multiple census repetitions.

6.2.2 Machine vision censuses in the greater context

To end this thesis, machine vision-assisted wildlife conservation needs to be put in context with surrounding disciplines. From a data source perspective, this for example includes research on UAV technology; in terms of products emerging from machine vision, the primary focus lies on conservation ecology; and beyond the direct protection of populations, a third perspective includes behavioral ecology. These three areas of research are shown in Figure 6.4 and stand for disciplines that are in direct contact with machine vision. The topic of wildlife censuses also affects more distant topics, such as international trade of horns and ivory (Biggs et al., 2013), but only in an indirect way. This last section briefly addresses research potential for those areas that can more directly be accelerated through machine vision.

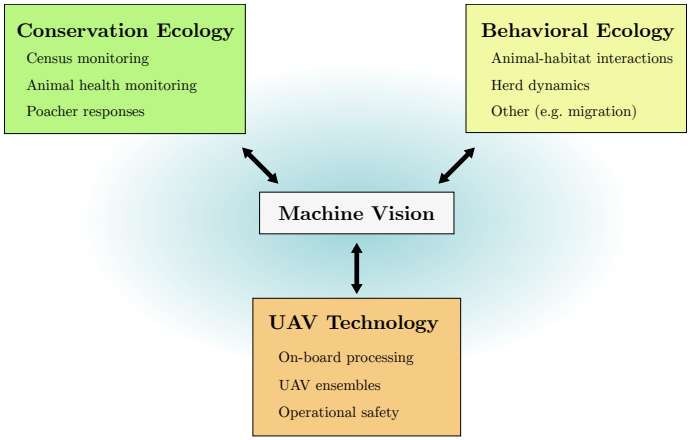


Figure 6.4: In the greater context, machine vision not only has the potential to accelerate conservation-related issues, but also to foster exciting new research frontiers, *e.g.* in behavioral ecology, or with respect to UAV technology from a more technical point of view.

Conservation ecology

Animal censuses and monitoring over time can be seen as one of the most direct products in terms of conservation ecology. However, the total abundance of a species in an area is just one aspect of conservation; another is the physical health of the individuals. This may include general physiological parameters like body temperature and heart rate (Kumar and Hancke, 2014), but also injuries and other incidental events on animals. It is unclear how and whether parameters like those can be monitored using CV, but accidents that *e.g.* leave visible scars on an individual's skin could be remotely observed and monitored

over time. For example, the wound of an injured animal may be treated initially, but the animal cannot be kept under surveillance until recovery. In these cases, a (semi-) automated, UAV-based recognition system may be helpful. From CV, this would require individual identification, and perhaps segmentation.

However, animal conservation at some point inevitably reaches a stage where passive methods, such as animal censuses, do not suffice anymore. In some extreme cases it may be required to actively target poachers directly. To this end, some research has been devoted to not only detect poachers, but also predict their next move in order to be able to prevent animal fatalities in the first place. This is commonly done using game theory or agent-based modeling (Nguyen et al., 2016). These models rely on sufficient amounts of training data like tracked positions of poachers throughout time, which again needs to be obtained using machine vision in the first place.

Behavioral ecology

There is a large amount of potential in employing machine vision wildlife monitoring systems for behavioral ecology. A majority of this research branch relies on dynamic interactions of individuals with other animals, or else their environment. For example, ecologists may be interested in interactions between individual animals and their herd, such as mother-offspring relationships (Murdock et al., 1983). Another behavioral trait that is highly relevant for African wildlife conservation is animal responses to humans, such as poachers (Happold, 1995). A third branch of research is the interaction of animals with their habitat: for example, correlating an animal or herd's preferred roaming or grazing area with the habitat type in terms of land cover could shed light on the requirements for the respective species, and on improvements for the animals' living conditions (Morrison et al., 2012). Edge effects of transitions between habitat types could be studied in similar ways (Porensky, 2011). Besides abundance, position, and trajectory estimations of animals, these applications also require land cover information that highlights the surface materials, and with that habitat types, of a given area. These in turn can also be obtained in large quantities using machine vision, through the form of semantic segmentation maps.

In return, behavioral studies of animals and herds may provide opportunities to further improve the prediction quality of machine vision models, through the form of *prior knowledge*. For example, a species' preference of habitat type or fodder plants may be indicative to where individuals are to be found in a wildlife area. Any potentially incorrect species predictions of CV models that happen to lie in less likely environments could be discarded this way. In terms of herd behavior, knowledge of the general herd size and density of a species can be exploited to *e.g.* recover missed animals in proximity of a detection, or to condition a detector in the first place. For example, the model responsible for the bird

predictions shown in Figure 6.2 managed to localize such a high number of birds precisely because it was conditioned with a prior, based on the expected flock density per species. This allowed estimating a certain number of expected individuals per area and species, and consequently required significantly less annotation efforts in the first place.

UAV technology

Finally, further developments for machine vision may also target the primary data source, *i.e.* UAVs and their acquisition modes. In conventional settings, UAVs are often programmed to fly over a transect in a regular pattern, *e.g.* lawnmower-like swipes across the area. This may be sufficient for censuses, but it falls short if other outputs are required. For example, in the case of individual animal tracking, one might want the UAV to automatically follow the individual and re-center its camera on it for as long as possible. This can only be achieved if either the UAV provides a (near) real-time data connection to a processing machine on the ground, or if the aircraft itself can do the calculations. The latter case is known as "on-board processing" and is these days being evaluated using down-sized, integrated platforms (Hulens et al., 2015). Works have shown that it is possible to perform object tracking on board a UAV (de Smedt et al., 2015), but the applicability to CV-based wildlife tracking still needs to be evaluated further.

6.3 Will we still have a need for interactive systems in the future?

At some point, the question arises whether UAVs with on-board processing will eventually become sophisticated enough to be completely autonomous. The theoretical benefits of such systems for wildlife conservation would be enormous: fleets of UAVs could be deployed to patrol the environs of a wildlife park as a safe alternative to human rangers, they could periodically assess the abundance, location, and health status of wildlife, and further alarm authorities in case of events that need human intervention. Given recent technological developments, the possibility of UAVs becoming autonomous might not even be that far in the future. Naturally, the aspects of ML, as well as Artificial Intelligence (AI), will play a major role to make this happen. Once autonomous systems have matured enough, census environments like the one introduced in this thesis are at the risk of being phased out and replaced completely—since they are interactive by design, they are not compatible with autonomous solutions after all. This inevitably raises the question of whether the work presented in this thesis is still going to be relevant in the future.

There are one or two highly critical requirements that need to be fulfilled, should we ever opt for autonomous systems. The first one is that they need to be near *failure-free*. In the

context of interactive systems, mistakes in predictions by the underlying models are not critical; in the worst case they may result in one or two misdetected animals that have to be re-annotated by humans. Importantly in this context, it is humans who have the last word in the decision process, not a machine. For autonomous systems, however, this does not apply. Human corrections are not available, and model mistakes may propagate infinitely, if they occur. Even worse, on board UAVs, ML models not only have to localize animals, but also need to be able to navigate the aircraft to follow the animals, stay within park boundaries, avoid obstacles, and more. All of these tasks involve physical interaction with the environment, and failures in the algorithms performing them could result in loss of equipment, or even worse, injuries to the animals.

However, the implications of autonomous systems reach further than just model mistakes. As a second requirement, these systems need to be applicable in an *ethical* manner. The present time sees rising concerns about AI and ethics, for example stemming from the fear of AI replacing human workforce without offering sufficient and adequate alternative jobs for them. Wildlife conservation is still more of a niche effort rather than a large-scale incentive, and the risk of park rangers becoming unemployed due to AI is probably not going to become critical. Rather, ethical concerns in conservation affect the threat of *poaching*. It is comparably hard for poachers to map a wildlife park with UAVs and perform photo-interpretation to localize and attack animals in real time, even if they do so with interactive machine vision assistance. But autonomous UAVs that provide real-time detection of animals could easily be abused for animal hunting, if put in the wrong hands. In such a scenario, authorities might respond with UAVs that are also armed, in order to physically interfere with poachers and prevent them from attacking. This is even more unethical and would drive the requirement for failure-free AI algorithms to the absolute limits, in order to ensure no harm is done to animals and civilians.

It is imperative for all of these problems to be addressed appropriately, should autonomous systems ever become an option for wildlife conservation. The scope of the challenges involved clearly reaches beyond prediction performances of AI algorithms onboard the autonomous systems themselves. Whether or not these challenges can be addressed sufficiently is still to be seen.

At this very moment, worldwide species decline is so severe that we cannot wait for autonomous systems to become available. Until then, and perhaps beyond, I therefore believe that interactive, machine vision-assisted systems, as a small link in a big chain in the fight against species and biodiversity loss, are the most appropriate way to go.

References

Andrew, W., Greatwood, C., and Burghardt, T. (2017). Visual localisation and individual identification of Holstein Friesian cattle via deep learning. In *IEEE International Conference on Computer Vision (ICCV)*.

Arteta, C., Lempitsky, V., and Zisserman, A. (2016). Counting in the wild. In *European Conference on Computer Vision (ECCV)*.

Attari, N., Ofli, F., Awad, M., Lucas, J., and Chawla, S. (2016). Nazr-CNN: object detection and fine-grained classification in crowdsourced UAV images. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*.

Baxter, P. W. and Hamilton, G. (2018). Learning to fly: integrating spatial ecology with unmanned aerial vehicle surveys. *Ecosphere*, 9(4).

Bayliss, P. and Yeomans, K. (1989). Distribution and abundance of feral livestock in the 'top end' of the Northern Territory (1985-86), and their relation to population control. *Wildlife Research*, 16(6):651–676.

Bazi, Y. and Melgani, F. (2018). Convolutional SVM networks for object detection in UAV imagery. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 56(6):3107–3118.

Bazzani, L., Bergamo, A., Anguelov, D., and Torresani, L. (2016). Self-taught object localization with deep networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L. (2016). What's the point: semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*.

Beery, S., van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *European Conference on Computer Vision (ECCV)*.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *26th Annual International Conference on Machine Learning (ICML)*.

Berger-Tal, O. and Saltz, D. (2014). Using the movement patterns of reintroduced animals to improve reintroduction success. *Current Zoology*, 60(4):515–526.

Biggs, D., Courchamp, F., Martin, R., and Possingham, H. P. (2013). Legal trade of Africa's rhino horns. *Science*, 339(6123):1038–1039.

Bilen, H., Pedersoli, M., and Tuytelaars, T. (2015). Weakly supervised object detection with convex clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blake, S., Douglas-Hamilton, I., and Karesh, W. (2001). Gps telemetry of forest elephants in central Africa: results of a preliminary study. *African Journal of Ecology*, 39(2):178–186.

Bondi, E., Dey, D., Kapoor, A., Piavis, J., Shah, S., Fang, F., Dilkina, B., Hannaford, R., Iyer, A., Joppa, L., and Tambe, M. (2018a). AirSim-W: a simulation environment for wildlife conservation with UAVs. In *1st ACM Conference on Computing and Sustainable Societies (COMPASS)*.

Bondi, E., Fang, F., Hamilton, M., Kar, D., Dmello, D., Choi, J., Hannaford, R., Iyer, A., Joppa, L., Tambe, M., et al. (2018b). Spot poachers in action: augmenting conservation drones with automatic detection in near real time. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Bouché, P., Lejeune, P., and Vermeulen, C. (2012). How to count elephants in west African savannahs? synthesis and comparison of main gamecount methods. *Biotechnologie, Agronomie, Société et Environnement*, 16(1):77–91.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bubnicki, J. W., Churski, M., and Kuijper, D. P. (2016). Trapper: an open source web-based application to manage camera trapping projects. *Methods in Ecology and Evolution*, 7(10):1209–1216.

Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Cai, W., Zhang, Y., and Zhou, J. (2013). Maximizing expected model change for active learning in regression. In *IEEE International Conference on Data Mining (ICDM)*.

Castelluccio, M., Poggi, G., Sansone, C., and Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*, 1508:1–11.

Caughley, G. (1974). Bias in aerial survey. *The Journal of Wildlife Management*, pages 921–933.

Ceballos, G., Ehrlich, P. R., Barnosky, A. D., García, A., Pringle, R. M., and Palmer, T. M. (2015). Accelerated modern human–induced species losses: entering the sixth mass extinction. *Science advances*, 1(5):e1400253.

Chamoso, P., Raveane, W., Parra, V., and González, A. (2014). UAVs applied to the counting and monitoring of animals. In *Ambient Intelligence-Software and Applications*, pages 71–80. Springer.

Cheng, G. and Han, J. (2016). A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11–28.

Collen, B., McRae, L., Deinet, S., de Palma, A., Carranza, T., Cooper, N., Loh, J., and Baillie, J. E. (2011). Predicting how populations decline to extinction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1577):2577–2586.

Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine Learning (ML)*, 20(3):273–297.

Côté, S. D. (1996). Helicopter disturbance. *Wildlife Society Bulletin*, 24(4):681–685.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(9):1853–1865.

Crall, J. P., Stewart, C. V., Berger-Wolf, T. Y., Rubenstein, D. I., and Sundaresan, S. R. (2013). Hotspotter—patterned species instance recognition. In *IEEE Workshop on Applications of Computer Vision (WACV)*.

Cuturi, M. (2013). Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS)*.

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN: object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Damodaran, B. B., **Kellenberger, B.**, Flamary, R., Tuia, D., and Courty, N. (2018). DeepJDOT: deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*.

de Kort, D., Altrichter, M., Cortez, S., and Camino, M. (2018). Collared peccary (Pecari tajacu) behavioral reactions toward a dead member of the herd. *Ethology*, 124(2):131–134.

de Smedt, F., Hulens, D., and Goedemé, T. (2015). On-board real-time tracking of pedestrians on a uav. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRw)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: a large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Després-Einspenner, M.-L., Howe, E. J., Drapeau, P., and Kühl, H. S. (2017). An empirical evaluation of camera trapping and spatially explicit capture-recapture models for estimating chimpanzee density. *American Journal of Primatology*, 79(7):e22647.

Díaz-Delgado, R., Mañez, M., Martínez, A., Canal, D., Ferrer, M., and Aragonés, D. (2017). Using UAVs to map aquatic bird colonies. In *The Roles of Remote Sensing in Nature Conservation*, pages 277–291. Springer.

Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., and Tian, Q. (2018). The unmanned aerial vehicle benchmark: object detection and tracking. In *European Conference on Computer Vision (ECCV)*.

Everingham, M., van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338.

Fang, Y., Du, S., Abdoola, R., Djouani, K., and Richards, C. (2016). Motion based animal detection in aerial videos. *Procedia Computer Science*, 92:13–17.

Ferreira, S. M. and Greaver, C. (2016). Re-introduction success of black rhinoceros in Marakele national park. *African Journal of Wildlife Research*, 46(2):135–138.

Ferreira, S. M., Greaver, C., Knight, G. A., Knight, M. H., Smit, I. P., and Pienaar, D. (2015). Disruption of rhino demography by poachers may lead to population declines in Kruger national park, South Africa. *PLoS One*, 10(6):e0127783.

Foglio, M., Semeria, L., Muscioni, G., Pressiani, R., and Berger-Wolf, T. (2019). Animal wildlife population estimation using social media images collections. *arXiv preprint arXiv:1908.01875*.

Gadiye, D., Eshiamwatta, G. W., and Odadi, W. O. (2016). Spatial-temporal distribution of the black rhino population in the Ngorongoro crater, Tanzania. *International Journal of Biological Research*, 4(2):232–236.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data. In *34th International Conference on Machine Learning (ICML)*, volume 70.

Gao, M., Li, A., Yu, R., Morariu, V. I., and Davis, L. S. (2018). C-WSL: count-guided weakly supervised localization. In *European Conference on Computer Vision (ECCV)*.

Girshick, R. (2015). Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Haalck, L., Mangan, M., Webb, B., and Risse, B. (2020). Towards image-based animal

tracking in natural environments using a freely moving camera. *Journal of Neuroscience Methods*, 330:108455.

Happold, D. (1995). The interactions between humans and mammals in africa in relation to conservation: a review. *Biodiversity & Conservation*, 4(4):395–414.

Harvey, R., Alden, C., and Wu, Y.-S. (2017). Speculating a fire sale: options for Chinese authorities in implementing a domestic ivory trade ban. *Ecological economics*, 141:22–31.

He, K. and Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hodgson, A., Kelly, N., and Peel, D. (2013). Unmanned aerial vehicles (UAVs) for surveying marine fauna: a dugong case study. *PLoS One*, 8(11):1–15.

Hodgson, J. C., Baylis, S. M., Mott, R., Herrod, A., and Clarke, R. H. (2016). Precision wildlife monitoring using unmanned aerial vehicles. *Scientific Reports*, 6:22574.

Hodgson, J. C., Mott, R., Baylis, S. M., Pham, T. T., Wotherspoon, S., Kilpatrick, A. D., Raja Segaran, R., Reid, I., Terauds, A., and Koh, L. P. (2018). Drones count wildlife more accurately and precisely than humans. *Methods in Ecology and Evolution*, 9(5):1160–1167.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. (2018). CyCADA: cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*.

Hollings, T., Burgman, M., van Andel, M., Gilbert, M., Robinson, T., Robinson, A., and McPherson, J. (2018). How do you find the green sheep? a critical review of the use of remotely sensed imagery to detect and count animals. *Methods in Ecology and Evolution*, 9(4):881–892.

Hulens, D., Goedemé, T., and Verbeke, J. (2015). How to choose the best embedded processing platform for on-board uav image processing? In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*.

Idrees, H., Tayyab, M., Athrey, K., and Zhang, D. (2018). Composition loss for counting, density map estimation and localization in dense crowds. In *European Conference on Computer Vision (ECCV)*.

Jachmann, H. (1991). Evaluation of four survey methods for estimating elephant densities. *African Journal of Ecology*, 29(3):188–195.

Kang, D., Ma, Z., and Chan, A. B. (2018). Beyond counting: comparisons of density maps for crowd analysis tasks – counting, detection, and tracking. *IEEE Transactions*

on Circuits and Systems for Video Technology, 29(5):1408–1422.

Kao, C.-C., Lee, T.-Y., Sen, P., and Liu, M.-Y. (2018). Localization-aware active learning for object detection. In *Asian Conference on Computer Vision (ACCV)*.

Kellenberger, B., Marcos, D., Courty, N., and Tuia, D. (2018a). Detecting animals in repeated UAV image acquisitions by matching CNN activations with optimal transport. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Kellenberger, B., Marcos, D., Lobry, S., and Tuia, D. (2019a). Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 57(12):9524–9533.

Kellenberger, B., Marcos, D., and Tuia, D. (2018b). Best practices to train deep models on imbalanced datasets—a case study on animal detection in aerial imagery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*.

Kellenberger, B., Marcos, D., and Tuia, D. (2018c). Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153.

Kellenberger, B., Marcos, D., and Tuia, D. (2019b). When a few clicks make all the difference: improving weakly-supervised wildlife detection in UAV images. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRw)*.

Kellenberger, B., Tuia, D., and Morris, D. (in revision). AIDE: accelerating image-based ecological surveys with artificial intelligence. *Methods in Ecology and Evolution*.

Kellenberger, B., Volpi, M., and Tuia, D. (2017a). Fast animal detection in UAV images using convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Kellenberger, B., Volpi, M., and Tuia, D. (2017b). Learning class-and location-specific priors for urban semantic labeling with CNNs. In *2017 Joint Urban Remote Sensing Event (JURSE)*.

Kideghesho, J. R. (2009). The potentials of traditional African cultural practices in mitigating overexploitation of wildlife species and habitat loss: experience of Tanzania. *International Journal of Biodiversity Science & Management*, 5(2):83–94.

Kingma, D. and Ba, J. (2014). Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Krishnappa, Y. S. and Turner, W. C. (2014). Software for minimalistic data management in large camera trap studies. *Ecological Informatics*, 24:11–16.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al. (2018). The sixth visual object

tracking VOT2018 challenge results. In *European Conference on Computer Vision (ECCV)*.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., and Pflugfelder, R. (2015). The visual object tracking VOT2015 challenge results. In *IEEE International Conference on Computer Vision workshops (ICCVw)*.

Krizhevsky, A., Sulskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Kumar, A. and Hancke, G. P. (2014). A zigbee-based animal health monitoring system. *Sensors Journal*, 15(1):610–617.

Lahiri, M., Tantipathananandh, C., Warungu, R., Rubenstein, D. I., and Berger-Wolf, T. Y. (2011). Biometric animal databases from field photographs: identification of individual zebra in the wild. In *1st ACM international conference on multimedia retrieval*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lewis, J. C. (1970). Wildlife census methods: a resume. *Journal of wildlife diseases*, 6(4):356–364.

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*.

Linchant, J., Lisein, J., Semeki, J., Lejeune, P., and Vermeulen, C. (2015). Are unmanned aircraft systems (UASs) the future of wildlife monitoring? A review of accomplishments and challenges. *Mammal Review*, 45(4):239–252.

Liu, J., Gao, C., Meng, D., and Hauptmann, A. G. (2017). DecideNet: counting varying density crowds through attention guided detection and density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: single shot multibox detector. In *European Conference on Computer Vision (ECCV)*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*.

Luo, T., Kramer, K., Goldgof, D. B., Hall, L. O., Samson, S., Remsen, A., and Hopkins, T. (2005). Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research (JMLR)*, 6:589–613.

Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-SVMs for object detection and beyond. In *IEEE International Conference on Computer Vision (ICCV)*.

Marcos, D., **Kellenberger, B.**, Volpi, M., and Tuia, D. (2018a). Scale equivariance in CNNs with vector fields. In *International Conference on Machine Learning workshops (ICMLw)*.

Marcos, D., Tuia, D., **Kellenberger, B.**, Zhang, L., Bai, M., Liao, R., and Urtasun, R. (2018b). Learning deep structured active contours end-to-end. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marcos, D., Volpi, M., **Kellenberger, B.**, and Tuia, D. (2018c). Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:96–107.

Meek, P. D., Ballard, G.-A., and Fleming, P. J. (2015). The pitfalls of wildlife camera trapping as a survey tool in Australia. *Australian Mammalogy*, 37(1):13–22.

Morrison, M. L., Marcot, B., and Mannan, W. (2012). *Wildlife-habitat relationships: concepts and applications*. Island Press.

Murdock, G. K., Stine, W. W., and Maple, T. L. (1983). Observations of maternal-infant interactions in a captive herd of sable antelope (hippotragus niger). *Zoo Biology*, 2(3):215–224.

Nazir, S., Newey, S., Irvine, R. J., Verdicchio, F., Davidson, P., Fairhurst, G., and van der Wal, R. (2017). WiseEye: next generation expandable and programmable camera trap platform for wildlife research. *PloS one*, 12(1):e0169758.

Nguyen, T. H., Sinha, A., Gholami, S., Plumptre, A., Joppa, L., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Critchlow, R., et al. (2016). Capture: a new predictive anti-poaching tool for wildlife protection. In *2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Niedballa, J., Sollmann, R., Courtiol, A., and Wilting, A. (2016). camtrapR: an R package for efficient camera trap data management. *Methods in Ecology and Evolution*, 7(12):1457–1462.

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., and Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *National Academy of Sciences*, 115(25):E5716–E5725.

Norton-Griffiths, M. (1978). *Counting animals*. Serengeti Ecological Monitoring Programme, African Wildlife Leadership Foundation.

Nowak, M. M., Dziób, K., and Bogawski, P. (2019). Unmanned aerial vehicles (UAVs) in environmental biology: a review. *European Journal of Ecology*, 4(2):56–74.

Ofli, F., Meier, P., Imran, M., Castillo, C., Tuia, D., Rey, N., Briant, J., Millet, P., Reinhard, F., Parkan, M., and Joost, S. (2016). Combining human computing and

machine learning to make sense of big (aerial) data for disaster response. *Big Data*, 4(1):47–59.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free? – weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. (2016). We don't need no bounding-boxes: training object class detectors using only human verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. (2017a). Extreme clicking for efficient object annotation. In *IEEE International Conference on Computer Vision (ICCV)*.

Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. (2017b). Training object class detectors with click supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Patel, V. M., Gopalan, R., Li, R., and Chellappa, R. (2015). Visual domain adaptation: a survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69.

Patton, F. and Campbell, P. (2011). Using eye and profile wrinkles to identify individual white rhinos. *Pachyderm*, pages 84–86.

Pelletier, C., Valero, S., Inglada, J., Champion, N., and Dedieu, G. (2016). Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 187:156–168.

Piel, A. K., Lenoel, A., Johnson, C. L., and Stewart, F. A. (2015). Deterring poaching in western Tanzania: the presence of wildlife researchers. *Global Ecology and Conservation*, 3:188–199.

Porensky, L. M. (2011). When edges meet: interacting edge effects in an African savanna. *Journal of Ecology*, 99(4):923–934.

Powers, R. P. and Jetz, W. (2019). Global habitat loss and extinction risk of terrestrial vertebrates under future land-use-change scenarios. *Nature Climate Change*, 9(4):323.

Qiao, Y., Truman, M., and Sukkarieh, S. (2019). Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Computers and Electronics in Agriculture*, 165:104958.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J. and Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Rey, N., Volpi, M., Joost, S., and Tuia, D. (2017). Detecting animals in African savanna with UAVs and the crowds. *Remote Sensing of Environment*, 200:341–351.

Risse, B., Mangan, M., Del Pero, L., and Webb, B. (2017). Visual tracking of small animals in cluttered natural environments using a freely moving camera. In *IEEE International Conference on Computer Vision (ICCV)*.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Ryan, D., Denman, S., Sridharan, S., and Fookes, C. (2015). An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding (CVIU)*, 130:1–17.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*.

Sasse, D. B. (2003). Job-related mortality of wildlife workers in the United States, 1937-2000. *Wildlife society bulletin*, pages 1015–1020.

Schlossberg, S., Chase, M. J., and Griffin, C. R. (2016). Testing the accuracy of aerial surveys for large mammals: an experiment with African savanna elephants (Loxodonta africana). *PloS one*, 11(10):e0164904.

Schlossberg, S., Chase, M. J., and Sutcliffe, R. (2019). Evidence of a growing elephant poaching problem in Botswana. *Current Biology*.

Schneider, S., Taylor, G. W., Linquist, S., and Kremer, S. C. (2019). Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods in Ecology and Evolution*, 10(4):461–470.

Schohn, G. and Cohn, D. (2000). Less is more: active learning with support vector machines. In *International Conference on Machine Learning (ICML)*.

Servick, K. (2014). Eavesdropping on ecosystems. *Science*, 343(6173):834–837.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and*

Machine Learning, 6(1):1–114.

Shang, C., Ai, H., and Bai, B. (2016). End-to-end crowd counting via joint learning local and global count. In *IEEE International Conference on Image Processing (ICIP)*.

Shi, M., Caesar, H., and Ferrari, V. (2017). Weakly supervised object localization using things and stuff transfer. In *IEEE International Conference on Computer Vision (ICCV)*.

Shi, M. and Ferrari, V. (2016). Weakly supervised object localization using size estimates. In *European Conference on Computer Vision (ECCV)*.

Shrivastava, A., Gupta, A., and Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Silveira, L., Jacomo, A. T., and Diniz-Filho, J. A. F. (2003). Camera trap, line transect census and track surveys: a comparative evaluation. *Biological conservation*, 114(3):351–355.

Silver, S. C., Ostro, L. E., Marsh, L. K., Maffei, L., Noss, A. J., Kelly, M. J., Wallace, R. B., Gómez, H., and Ayala, G. (2004). The use of camera traps for estimating jaguar Panthera onca abundance and density using capture/recapture analysis. *Oryx*, 38(2):148–154.

Sindagi, V. A. and Patel, V. M. (2018). A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16.

Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958.

Srivastava, S., Berman, M., Blaschko, M. B., and Tuia, D. (2019). Adaptive compression-based lifelong learning. In *British Machine Vision Conference (BMVC)*.

Stark, D. J., Vaughan, I. P., Evans, L. J., Kler, H., and Goossens, B. (2018). Combining drones and satellite tracking as an effective tool for informing policy change in riparian habitats: a proboscis monkey case study. *Remote Sensing in Ecology and Conservation*, 4(1):44–52.

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., and Packer, C. (2015). Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, 2:150026.

Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., Vercauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., Teton, B., Beasley, J. C., Schlichting, P. E., Boughton, R. K., Wight, B., Newkirk, E. S., Ivan, J. S., Odell, E. A., Brook, R. K., Lukacs, P. M., Moeller, A. K., Mandeville, E. G., Clune, J., and Miller, R. S. (2019). Machine learning to classify animal species in camera trap images: applications in ecology. *Methods in Ecology and Evolution*, 10(4):585–590.

Tuia, D. and Camps-Valls, G. (2016). Kernel manifold alignment for domain adaptation. *PloS one*, 11(2):e0148655.

Tuia, D., **Kellenberger, B.**, Pérez-Suey, A., and Camps-Valls, G. (2018). A deep network approach to multitemporal cloud detection. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Tuia, D. and Muñoz-Marí, J. (2013). Learning user's confidence for active learning. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 51(2):872–880.

Tuia, D., Pasolli, E., and Emery, W. J. (2011a). Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242.

Tuia, D., Persello, C., and Bruzzone, L. (2016). Domain adaptation for the classification of remote sensing data: an overview of recent advances. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):41–57.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., and Muñoz-Marí, J. (2011b). A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617.

Uijlings, J. R., van de Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171.

Ulyanov, D. and Vedaldi, A. (2016). Instance normalization: the missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022v3*.

van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9:2579–2605.

van Gemert, J. C., Verschoor, C. R., Mettes, P., Epema, K., Koh, L. P., and Wich, S. (2014). Nature conservation drones for automatic localization and counting of animals. In *European Conference on Computer Vision (ECCV)*.

Wang, L., Shao, W., Lu, Y., Ye, H., Pu, J., and Zheng, Y. (2018). Crowd counting with density adaption networks. *arXiv preprint arXiv:1806.10040*.

Wasser, S. K. and Gobush, K. S. (2019). Conservation: monitoring elephant poaching to

prevent a population crash. *Current Biology*, 29(13):R627–R630.

Weinstein, B. G. (2015). MotionMeerkat: integrating motion video detection and ecological monitoring. *Methods in Ecology and Evolution*, 6(3):357–362.

Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., Veldthuis, M., and Fortson, L. (2019). Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, 10(1):80–91.

Witter, R. and Satterfield, T. (2019). Rhino poaching and the "slow violence" of conservation-related resettlement in Mozambique's Limpopo national park. *Geoforum*, 101:275–284.

Wu, Y., Lim, J., and Yang, M.-H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1834–1848.

Xue, Y., Wang, T., and Skidmore, A. K. (2017). Automatic counting of large mammals from very high resolution panchromatic satellite imagery. *Remote Sensing*, 9(9):878.

Yang, Z., Wang, T., Skidmore, A. K., de Leeuw, J., Said, M. Y., and Freer, J. (2014). Spotting east african mammals in open savannah from space. *PLoS One*, 9(12):e115989.

Zhang, C., Li, H., Wang, X., and Yang, X. (2015). Cross-scene crowd counting via deep convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Y., Zhou, D., Chen, S., Gao, S., and Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. (2015). Scalable person re-identification: a benchmark. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017). Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: locating object proposals from edges. In *European Conference on Computer Vision (ECCV)*.

Acknowledgments

The proposition of this thesis to put humans in an interactive loop is not the product of chance—it is through extensive interaction, and indispensable support by fellows, colleagues, and relatives, that I could even reach the stage of writing this manuscript. First and foremost, my greatest thanks go to my supervisor Devis Tuia, who put me in the most productive loop with enthusiastic support whenever possible, and effectively pulled me out of unproductive loops whenever I was at my wits' end. We conceptualized, shaped, iterated, and improved not only this work together, but everything beyond the inner scientific circle, right from the start of this project when we were still at the University of Zurich.

To continue the round, I would like to express my gratitude to my colleagues Diego Marcos, Sylvain Lobry, and Michele Volpi, who all provided me with essential inputs and support in all of the projects' discussion cycles, and great company throughout the journey. One never has enough eyes to see all facets of a project, and having support in the form I enjoyed was a major pillar in the realization of all of my work during the iteration of my PhD.

Beyond the circumference of the research group I would like to thank Dan Morris at Microsoft AI for Earth for hosting and supporting me during Summer 2019. The internship was an exceptional opportunity and allowed me to make the final step from scientific findings to impact.

Finally, to close the round, my thanks and best wishes go to everyone who enriches my life by interaction, support, and presence, including my fellow members of the Laboratory of Geo-Information Science and Remote Sensing in Wageningen, the Remote Sensing Laboratories in Zurich and, above all, my family, Michelle and Mathias Kneubühler, and Roman and Tobias Kellenberger.

About the author

Benjamin Kellenberger was born in Zurich, Switzerland on March 23, 1992. Always inclined towards technology, he nonetheless studied a BSc and MSc geography at the University of Zurich, albeit with specialization in Geo-Information Science and Remote Sensing, as well as with a minor in computer science. Excited by the possibilities of digital automation, he deepened his programming skills in two internships at the Federal Office of Topography swisstopo, and the Institute of Cartography and Geoinformation at ETH Zurich. Both employments resulted in software products that involved extensive interfacing between humans and computers, one of which was the GeoVITe geodata download portal[7]. In February 2016, he joined the research group of Devis Tuia, then at the University of Zurich, and commenced his PhD. This gave him the unique opportunity to combine interests (programming, interfacing) with desire (environment protection), while including exciting skills from Machine Learning and Computer Vision (CV). In 2017, he moved with the group to Wageningen University, the Netherlands, where he finished his PhD thesis and intends to build on the work through a postdoctoral position.

Peer-reviewed journal/CV conference publications

Kellenberger, B., Tuia, D., and Morris, D. (in revision). AIDE: accelerating image-based ecological surveys with artificial intelligence. *Methods in Ecology and Evolution*.

Kellenberger, B., Marcos, D., Lobry, S., and Tuia, D. (2019a). Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 57(12):9524–9533.

Kellenberger, B., Marcos, D., and Tuia, D. (2019b). When a few clicks make all the difference: improving weakly-supervised wildlife detection in UAV images. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRw)*.

Damodaran, B. B., **Kellenberger, B.**, Flamary, R., Tuia, D., and Courty, N. (2018). DeepJDOT: deep joint distribution optimal transport for unsupervised domain adapta-

[7] https://geovite.ethz.ch

tion. In *European Conference on Computer Vision (ECCV) (joint first author)*.

Marcos, D., **Kellenberger, B.**, Volpi, M., and Tuia, D. (2018a). Scale equivariance in CNNs with vector fields. In *International Conference on Machine Learning workshops (ICMLw)*.

Marcos, D., Tuia, D., **Kellenberger, B.**, Zhang, L., Bai, M., Liao, R., and Urtasun, R. (2018b). Learning deep structured active contours end-to-end. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marcos, D., Volpi, M., **Kellenberger, B.**, and Tuia, D. (2018c). Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:96–107.

Kellenberger, B., Marcos, D., and Tuia, D. (2018c). Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153.

Other scientific publications

Tuia, D., **Kellenberger, B.**, Pérez-Suey, A., and Camps-Valls, G. (2018). A deep network approach to multitemporal cloud detection. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Kellenberger, B., Marcos, D., Courty, N., and Tuia, D. (2018a). Detecting animals in repeated UAV image acquisitions by matching CNN activations with optimal transport. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Kellenberger, B., Marcos, D., and Tuia, D. (2018b). Best practices to train deep models on imbalanced datasets—a case study on animal detection in aerial imagery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*.

Kellenberger, B., Volpi, M., and Tuia, D. (2017a). Fast animal detection in UAV images using convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Kellenberger, B., Volpi, M., and Tuia, D. (2017b). Learning class-and location-specific priors for urban semantic labeling with CNNs. In *2017 Joint Urban Remote Sensing Event (JURSE)*.

PE&RC Training and Education Statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the C.T. de Wit Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 32 ECTS (= 22 weeks of activities)

Review of literature (6 ECTS)

- Machine vision for wildlife monitoring with Unmanned Aerial Vehicles (UAVs) and humans in the loop

Writing of project proposal (4.5 ECTS)

- Detecting and tracking objects in UAV videos

Post-graduate courses (2.5 ECTS)

- Vision and sports summer school; PhD summer school on computer vision science; Vittorio Ferrari, CTU Prague (2016)
- Principles and theory in geography; University of Zürich (2016)

Laboratory training and working visits (4.5 ECTS)

- Wildlife detection in aerial images; Microsoft AI for Earth (2019)

Invited review of (unpublished) journal manuscript (2 ECTS)

- IEEE TGRS: building extraction at scale using convolutional neural network: a case study in the United States (2017)
- IEEE TGRS: deep feature alignment neural networks for domain adaptation of hyperspectral data (2017)

Deficiency, refresh, brush-up courses (3 ECTS)

- Machine learning for geosciences; University of Zürich (2016)

Competence strengthening / skills courses (2 ECTS)

- Project management; University of Zürich (2016)
- PhD Seminar: doing a PhD; University of Zürich (2017)

PE&RC Annual meetings, seminars and the PE&RC weekend (1.2 ECTS)

- PE&RC Last year's weekend
- Department of Geography Graduate School retreat; Zürich

Discussion groups / local seminars / other scientific meetings (4.5 ECTS)

- Remote sensing seminars; Zurich (2015, 2016)
- NCG Symposium; TU Delft (2017)
- NCG Symposium; WUR (2018)

International symposia, workshops and conferences (9.4 ECTS)

- Joint Urban Remote Sensing Event; oral presentation; Dubai, UAE (2017)
- IGARSS Remote Sensing conference; oral presentation; Fort Worth, TX, USA (2017)
- IGARSS; oral presentation; Valencia, E (2018)
- ECML-PKDD Machine Learning conference; oral presentation, Dublin, I (2018)

Lecturing / supervision of practicals / tutorials (1 ECTS)

- Machine learning for geosciences (2017)

Supervision of MSc students (9 ECTS)

- Suitability assessment of object tracking for animals in UAV imagery
- Addressing biases in class imbalances for animal detection in camera trap images
- Using synthetic training data for animal detection in aerial imagery

Financial support from Wageningen University for printing this thesis is gratefully acknowledged.

Cover: Animal foregrounds based on pictures of the Kuzikus wildlife reserve (image courtesy of Friedrich Fedor Reinhard). Backdrops show the Orion constellation and nebula, as seen from the ground in Wageningen, January, 2020 (photographed by Benjamin Kellenberger). Front and back cover design by Benjamin Kellenberger.

Printed by ProefschriftMaken (http://www.proefschriftmaken.nl)