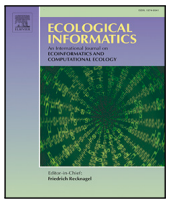
Ecological Informatics 91 (2025) 103387

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



How to minimize the annotation effort in aerial wildlife surveys

Giacomo May [a,*], Emanuele Dalsasso [a], Alexandre Delplanque [b], Benjamin Kellenberger [c,d], Devis Tuia [a]

[a] Environmental Computational Science and Earth Observation Laboratory, École Polytechnique Fédérale de Lausanne, Rte des Ronquos 86, Sion, 1950, Switzerland
[b] TERRA Teaching and Research Centre (Forest Is Life), ULiège, Gembloux Agro-Bio Tech, 2 Passage des Déportés, Gembloux, 5030, Belgium
[c] Centre for Biodiversity and Environmental Research, University College London, One Pool Street, London, E20 2AF, United Kingdom
[d] Department of Ecology and Evolutionary Biology, Yale University, Osborn Memorial Laboratories, 165 Prospect Street, New Haven, CT 06520-8105, United States of America

ARTICLE INFO

Dataset link: LILA BC, Zenodo, GitHub

Keywords:
Object detection
Object localization
Density estimation
Aerial wildlife censuses
Annotation effort
Pseudo labels
Point annotations

ABSTRACT

Aircraft-based monitoring of wildlife is a popular way among conservation practitioners to obtain animal population counts over large areas. Nowadays, these aerial censuses are becoming increasingly scalable due to the advent of drone technology, which is frequently combined with deep learning-based image recognition. Yet, the annotation burden associated with training deep learning architectures remains a problem especially for commonly used bounding box detection models. Point-based density estimation- and localization models are cheaper to train, and often work better when the aerial imagery is recorded at an oblique angle. Beyond this, though, there currently is little consensus about which strategy to use for what kind of data. In this work, we address this knowledge gap and evaluate modifications to a state-of-the-art detection model (YOLOv8) that minimize labeling efforts by enabling it to work on point-annotated images. We study the effect of these adjustments on detection accuracy and extensively compare them to a localization architecture on four datasets consisting of nadir and oblique images. The goal of this paper is to offer wildlife conservationists practical advice on which of the recently proposed deep learning architectures to use given the properties of their images, as well as on the data properties that will maximize model performance independently of the architecture. We find that counting accuracy can largely be maintained at reduced annotation effort, that object detection technology outperforms the localization approach on nadir images, and that it shows competitive performance in the oblique setting. The images used to obtain the results presented in this paper can be found on Zenodo for all publicly available datasets, as well as all code necessary to reproduce our results was uploaded to GitHub.

1. Introduction

Rapid loss of biodiversity across the globe (Ceballos et al., 2020) creates an urgent need for efficient and accurate monitoring of wildlife, where counting the number of animals in nature reserves is of particular importance (Jachmann, 2012). Such censuses are often performed from airplanes with human observers either directly spotting wildlife from the aircraft (Ottichilo et al., 2000; Stapleton et al., 2016; Michaud et al., 2014; Jackmann, 2001), and/or taking images for posterior analysis (Delplanque et al., 2023; Eikelboom et al., 2019). An increasingly wide-spread alternative to this approach is the usage of drones (Seymour et al., 2017; Kellenberger et al., 2017; Lyons et al., 2019; Kellenberger et al., 2018; Chen et al., 2023a; Rey et al., 2017), which – at the cost of reduced range – are safer and cheaper to operate than airplanes (Linchant et al., 2015) and can autonomously record images of their flights along swaths of the study area, allowing up to full coverage instead of only flight transects.

Counting animals in the acquired images is frequently done with the help of machine learning (Tuia et al., 2022), where especially deep neural networks are widely used (Kellenberger et al., 2017; Dujon et al., 2021; Rančić et al., 2023; Maire et al., 2015; Delplanque et al., 2022; Eikelboom et al., 2019; Peng et al., 2020; Delplanque et al., 2024; Xu et al., 2024) due to their excellent accuracy (Alzubaidi et al., 2021). Deep learning has also been applied to the identification of animals in satellite imagery (Borowicz et al., 2019; Wu et al., 2023; Duporge et al., 2021; Gonçalves et al., 2020). This approach greatly increases the spatial scale of wildlife surveys, but is confined to the detection of relatively large animals by image resolution. Furthermore, there is currently no evidence that satellites can be used to distinguish between

* Corresponding author.
E-mail address: giacomo.may@epfl.ch (G. May).

https://doi.org/10.1016/j.ecoinf.2025.103387
Received 23 February 2025; Received in revised form 8 August 2025; Accepted 11 August 2025
Available online 19 August 2025

different animal classes (Xu et al., 2024), preventing their application to multi-species censuses.

Among the deep learning architectures used for aerial wildlife monitoring, object detection models are the most common method (Xu et al., 2024), which output bounding boxes (rectangles) around identified animals, along with the name of their species. This creates two problems: first, the bounding box labels required to train these models are expensive to obtain. Second, it is unclear to which degree detection architectures are capable of dealing with occlusion and scale variation of objects in *oblique* images – i.e., aerial imagery captured with the camera axis inclined to one side instead of pointing straight to the ground (*nadir* setting). As oblique images are common in aerial wildlife observation (Delplanque et al., 2023), recent works have abandoned object detection in favor of density estimation- (Meena et al., 2023; Padubidri et al., 2021) and localization models (Delplanque et al., 2023), which promise increased robustness to said problems of occlusion and scale variation. Instead of predicting a set of bounding boxes, these architectures generate 2-D maps whose pixel values encode either the estimated density of Jiang et al. (2019), Liu et al. (2019) or the distance to Arteta et al. (2016), Xu et al. (2022) objects at the underlying image location. The ground truth maps necessary to train these image-to-map (I2M) models can easily be obtained from point labels, which are up to seven times quicker to generate compared to bounding boxes (Ge et al., 2023), and thereby represent an additional advantage over the traditional box-based detection approach.

At the same time, solutions exist to reduce the annotation effort of state-of-the-art (SOTA) bounding-box object detectors to that of I2M models. Specifically, architectural modifications to ensure point compatibility of object detection models have been proposed (May et al., 2024), as well as *pseudo-boxes* can be used for training, which are created in automated fashion by drawing squares of predefined sizes around point annotations (Ribera et al., 2019; Yu et al., 2022). However, individuals of a certain species can vary in size (cf. Section 3.3), meaning that, in many cases, pseudo-boxes will either be larger or smaller than the actual specimen. Pseudo-labels thus provide inaccurate information about the dimensions of animals - a problem that is exacerbated when using point annotations, as they do not encode any spatial information at all. Points are hence considered less complex than pseudo-boxes, which in turn are less complex than hand-crafted boxes.

It yet remains unclear *(i)* how strongly these reductions in label complexity affect detection performance in the context of aerial wildlife censuses, and *(ii)* how object detection models compare to I2M methods at reduced labeling effort and under varying acquisition geometry (nadir vs. oblique).

We aim to bring an answer to these two questions by comparing and extensively evaluating recent point-based models with bounding box-based object detectors on a diverse set of aerial wildlife imagery. Specifically, we compare `POLO` (May et al., 2024), the point-compatible version of the `YOLOv8` framework, `YOLOv8_p`, a regular `YOLOv8` trained on pseudo-boxes, and `HerdNet`, an I2M model specifically developed for aerial surveys of animal herds (Delplanque et al., 2023) on a total of four different datasets (two oblique, and two nadir image collections). We moreover compare the two aforementioned `YOLOv8` modifications to `YOLOv8` models trained on hand-crafted bounding boxes, and introduce a set of novel metrics for the evaluation of point-based detectors.

This work is intended to provide guidance to ecologists and conservation practitioners who wish to leverage existing deep learning techniques to automate the aerial monitoring of animal populations, but may be unsure about which architecture to choose and what kind of data to collect. By providing actionable solutions and recommendation, we aim to reduce the gap between methodological developments in aerial wildlife conservation and their practical implementation by wildlife conservation agencies and practitioners.

2. Related works

We focus on three approaches: object detection, I2M, and point-based object detection. Animal counting can be formulated as a regression problem (Norouzzadeh et al., 2018; Hoekendijk et al., 2021) as well, but we decided not to include such methods, since regression models predict only the total number of animals in an image, without specifying their location. The latter can, however, represent important information for end-users (Schneider et al., 2024). Moreover, while it suffices to provide animal counts for each image in order to train regression models - and it hence may be argued that the labeling burden is lower than that of point detection or I2M models -, it is likely to be similar (Lempitsky and Zisserman, 2010): Especially when pictures contain a high number of animals, (which is common in aerial imagery; cf. *Density* column of Table 1), annotators will have to mark the individuals already counted to keep track of their progress. Since this will (at the very least) involve clicking on every animal, the effort becomes equivalent to that of generating point labels.

Finally, the recent emergence of so called foundation models has attracted interest within the ecological community (Morera, 2024) as a means to drastically reduce or even remove the need for annotations. Foundation models are transformer-based architectures that exhibit strong generalization capacities through extensive pre-training on ultra-scale datasets, allowing researchers to apply them to their individual use-case with little or no additional labeling requirements. So far, these models have mainly been deployed for animal and behavioral classification in camera trap data (Fabian et al., 2023; Gabeff et al., 2024), though in a very recent work (Lalgudi et al., 2025) use a combination of CLIP (Radford et al., 2021) and the Segment Anything model (SAM) (Kirillov et al., 2023) for fully-autonomous zero-shot shark tracking from drone videos. In the future, a more widespread integration of these architectures into conservation practices, especially within the present context of aerial population monitoring, is likely but will require overcoming current failure cases, including SAM's lack of accuracy on overhead imagery, small or concealed objects, and blurry edges (Ren et al., 2024; Ji et al., 2023; Chen et al., 2023b).

2.1. Object detection

Object detection algorithms are traditionally divided into one- and two-stage architectures, where `Faster R-CNN` (two-stage; Girshick, 2015) and `YOLO` (one-stage; Redmon et al., 2016) are the two most popular choices for processing aerial imagery of wildlife (Xu et al., 2024). Broadly speaking, two-stage detectors first identify regions of an image that are likely to contain an object, and in a second step focus on these regions to predict the bounding boxes and object classes within them. `YOLO` bypasses the region proposal step and divides the input image into a grid of square cells, each one responsible for detecting objects that lie inside of it. Notably, models may fall outside of these two categories, as is the case for vision transformers (e.g., `DETR` Carion et al., 2020) who leverage various attention mechanisms to better capture long range dependencies and global context within an image.

`YOLO` is considered to be the SOTA solution for object detection, with recent versions consistently outperforming `Faster R-CNN` (Mou et al., 2023; Doll and Loos, 2023), while remaining competitive with transformer architectures (Jrondi et al., 2024; Wang et al., 0000; Rajput et al., 2024). In this work, we will thus focus on `YOLO` as a base architecture and use `v8` as the starting model.

Related efforts towards exploring the potential of the `YOLO` and `Faster R-CNN` frameworks for animal detection from aerial imagery have been undertaken by Ye et al. (2025), Naidu et al. (2025), Ascagorta et al. (2025), Desgarnier et al. (2022) and Sharma et al. (2018). The `ADD-YOLO` model developed by Ye et al. (2025) improves the network structure of `YOLOv8` through the addition of novel modules which improve detection accuracy on small and occluded animals. In a similar vein, Naidu et al. (2025) also add modules to `YOLO`,

and implement a novel loss function. They apply their solution to the detection of wild and domesticated ungulates. Ascagorta et al. (2025) use a YOLO model to detect and monitor populations of elephant seals and sea lions along the Valdés Peninsula of Argentina. Faster R-CNN models are employed by Desgarnier et al. (2022), and Sharma et al. (2018) for detecting sharks and eagle rays from drone imagery.

2.2. I2M

I2M approaches, as we call them here, were developed in the crowd counting literature to avoid issues of occlusion and scale variation by predicting density- or localization maps (Gao et al., 2003; Bai et al., 2022; Zhu et al., 2021). Zhu et al. (2021) extensively benchmark various density estimation architectures on a large, partly web-scraped dataset of drone images taken at different altitudes, camera angles, and containing varying degrees of animal density. They do not draw any comparisons between object detection and density estimation. Delplanque et al. (2023) close this gap by developing a localization model specifically for the purpose of counting dense animal herds photographed from the oblique perspective, and comparing it to an object detection and density estimation baseline. Yet, their efforts leave unanswered questions about how architectures compare in the nadir setting, which can be expected to become more and more common with the rise of drone technology in wildlife monitoring (Linchant et al., 2015). Also, the authors use Faster R-CNN as their object detection baseline, which – as reviewed above – is a less accurate choice with respect to recent versions of YOLO (Hong et al., 2019; Andrew et al., 2021; Bondi et al., 2020).

Finally, Laradji et al. (2018), and Arteta et al. (2016) use I2M models to count penguin colony numbers from images recorded by stationary cameras mounted slightly above ground level in Antarctica. In particular, Laradji et al. (2018) train their model to output binary classification maps from which "blobs" that roughly encapsulate the animals are derived, whereas Arteta et al. (2016) stick to predicting density maps but add the task of estimating a foreground–background segmentation.

2.3. Point-based object detection

P2PNet (Song et al., 2021) is one example for a point-based object detection framework, *i.e.*, a model that can be trained on (and output) point labels. It allows to go beyond simple counting by enabling precise localization of individuals in crowds. POLO (May et al., 2024) on the other hand leverages the well-established YOLO algorithm by building on the YOLOv8 architecture. It introduces simple, yet effective modifications to enable point-compatibility, while minimizing the differences to its bounding-box-based counterpart. This similarity in the models' design enables us to isolate the effect of reduced label complexity from other architectural confounding factors, making POLO an ideal fit for the purposes of this study.

3. Methods

We aim at comparing the suitability of object detection vs. I2M for aerial censuses of animal populations. To do so, we benchmark architectures based on bounding box detection, point detection, and localization on four different datasets. We choose YOLOv8 (Jocher et al., 2023) as the SOTA bounding box framework, POLO (May et al., 2024) as the point detection model, and HerdNet (Delplanque et al., 2023) as the I2M baseline, since it has proven superior to density estimation for the given task (Delplanque et al., 2023).

In what follows, we highlight the technical details underlying the architectures (Section 3.1), outline the metrics we compute to measure detection and counting accuracy – including those based on the *Distance over Radius* (DoR) metric we propose (Section 3.2) –, and introduce the datasets and hyperparameters used for training the models (Sections 3.3 and 3.4). Diagrams of all architectures are provided in Fig. 1 and Fig. 2.

3.1. Architectures

3.1.1. YOLOv8

The YOLOv8 architecture (Jocher et al., 2023) processes images by first passing them through a sequence of convolutions and pooling layers, also referred to as the encoder. Each convolution thereby generates a latent representation of the input that contains increasingly coarse and semantically meaningful features. The last of these feature maps, i.e., the final output of the encoder, is then sent to the model's head along with two representations produced by earlier convolutions. In this way, YOLOv8 leverages feature maps with varying levels of granularity and abstraction, which is useful for detecting objects of different scale and dimensions. The head processes each of the three representations via two separate branches, one specialized in localization (locating animals) and one in classification (predicting their species). Here, every branch uses an additional encoder, meaning another series of convolutions, to obtain a final set of feature maps. Both branches finally compute one prediction for every pixel of these feature maps, yielding one bounding box and one class estimate per feature map pixel. In most cases, this will be excessive. YOLOv8 hence removes redundant outputs by first filtering low-confidence detections, i.e., detections whose class probabilities fall below a certain threshold, and then removing bounding boxes with a high degree of overlap, as this is indicative of the same object being detected multiple times – a process called *non-maximum-suppression* (NMS).

Three different loss terms are employed during training:

1. **IoU loss:** This loss measures how well the model's predictions match the ground truth bounding boxes geometrically, where small overlap leads to increased penalization. As is the case during NMS, the amount of overlap is quantified through the *Intersection-over-Union* (IoU) metric, the intersection of two bounding boxes A and B divided by their union:

$$IoU = \frac{A \cap B}{A \cup B} \tag{1}$$

.

2. **DFL:** In synergy with the IoU loss, the *Distribution Focal Loss* (DFL) enforces learning the width and the height of bounding boxes. More precisely, it is responsible for learning a probability distribution for the offsets of bounding box edges from the center of feature map pixels the trained model can draw from at inference time. Here, distributions with sharp peaks indicate that the model has a clear grasp of the typical dimensions of objects, which will lead to tightly fitted boxes, and which therefore is what the DFL optimizes towards during training. (Li et al., 2020).
3. **BCE loss:** To penalize incorrect class predictions for the detected objects, YOLOv8 applies a *binary-cross-entropy* (BCE) loss. While the categorical cross entropy (CCE) would be the typical choice for classification problems that include multiple classes, employing the BCE allows the YOLOv8 framework to be used in multi-label situations – i.e., when a single object can belong to more than one category.

3.1.2. POLO

The POLO architecture, first proposed by May et al. (2024), is largely based on the YOLOv8 model, with notable exceptions. First, as POLO is designed to predict points instead of bounding boxes, the last feature map issued by the encoder of the localization branch contains only two channels. For each pixel of this feature map, the values in the first channel, a_1, serve to compute the x- ($\hat{p}_x$) coordinate of animals' center points, whereas the second channel values, a_2, are used to calculate the corresponding y-coordinate ($\hat{p}_y$). Specifically:

$$\begin{aligned} \hat{p}_x &= \sigma(a_1) \cdot 2 - 0.5 + c_x \\ \hat{p}_y &= \sigma(a_2) \cdot 2 - 0.5 + c_y \end{aligned} \tag{2}$$

, where $\sigma(\cdot)$ represents the sigmoid function. Object coordinates are predicted as offsets from the top-left corner of each feature map pixel, indicated by coordinates c_x and c_y. Importantly, possible offsets range between −0.5 and 1.5, thus allowing for predictions into neighboring regions, which helps to accurately locate objects that span multiple feature map pixels.

POLO furthermore replaces the IoU loss and DFL with the *mean squared error* (MSE) between an estimated point $\hat{p}_i$ and the corresponding label p_i:

$$MSE = \frac{1}{|P|}\sum_{i=1}^{|P|} \|p_i - \hat{p}_i\|_2^2 \tag{3}$$

, where P denotes the set of ground truth annotations. Analogously to YOLOv8, a BCE loss is used for species classification.

Lastly, to filter out geometrically redundant detections during NMS, POLO replaces the IoU with the DoR, which is defined as the Euclidean distance d between two points p_1 and p_2, divided by the radius value r_c assigned to the class (animal species) of p_1:

$$DoR = \frac{d(p_1, p_2)}{r_c} \tag{4}$$

Here, a low DoR value indicates close spatial proximity and thus redundancy, as points below a given DoR threshold are likely to lie on the same animal.

3.1.3. HerdNet

The task of predicting 2-D maps from input images is usually achieved by an encoder–decoder structure. Similarly to what is the case for YOLOv8 and POLO, the encoder extracts low-resolution, latent representations from input images. However, in I2M this feature map is projected back to the input space through a series of up-sampling operations, called the decoder. Throughout this process, feature maps from different layers of the encoder and decoder are concatenated to obtain latent representations that capture features of different spatial scales. HerdNet uses the DLA-34 encoder–decoder structure, which implements iterative and hierarchical aggregation strategies (Yu et al., 2018).

The output of the HerdNet decoder is passed through two more convolutional layers to obtain the final prediction in the form of a *Focal Inverse Distance Transform Map* (FIDTM) (Liang et al., 2022), where each cell in the prediction grid contains the expected inverse distance to the nearest object in the input image. A 3 × 3 max-pooling kernel is applied to the FIDTMs in order to identify local maxima and locate individual animals. The animals' species is then predicted by overlaying the FIDTMs with a low-resolution classification map, predicted directly from the encoder's output (cf. Fig. 2).

3.2. Model evaluation metrics

To measure the counting accuracy of our models, we calculate the *mean absolute error* (MAE) and *mean squared error* (MSE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |x_i - \hat{x}_i| \tag{5}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{6}$$

, where N is the number of images in the dataset, $\hat{x}_i$ is the number of animals detected in image i, and x_i is the ground truth count for i. The predicted and ground truth count can be species-specific or binary. In the latter case $\hat{x}_i$ and x_i refer to the total number of animals, regardless of their class. For example, the binary count for an image with a species-specific count of 3 zebras and 9 elephants would be 12.

3.2.1. Bounding box metrics

For bounding box models, the accuracy of predictions is assessed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Confusion = 1 - \frac{TP}{TP + FP_{fg}} \tag{9}$$

A prediction is considered a true positive (TP) if it assigns the correct species, and if the estimated bounding box exceeds a certain degree of overlap with a ground truth label, as measured through the IoU. Detections are classified as false positives (FP) if they cannot be matched to any ground truth (IoU with nearest ground truth is below threshold), if they match an animal that has already been matched by another, geometrically more accurate detection, or if they are geometrically correct, but assign the wrong class. We refer to the last case, where the wrong class is predicted, as a foreground false positive FP_{fg}. Lastly, if no correct prediction can be found for a ground truth object, it is classified as a false negative (FN). This means that an object of class A that is detected as an object of class B, counts as FP for class B and as a FN for class A.

As described in Section 3.1, YOLOv8 removes excess detections by filtering out predictions with probabilities below a certain threshold via NMS. Lowering this threshold will lead to less pruning and a reduction in the amount of false negatives, while increasing the number of false positives. Therefore, the probability threshold significantly influences the recall and precision performance of a model, which can be visualized by plotting recall values against precision scores at increasing threshold values. This yields the so-called *Precision-Recall-Curve* (PR-curve). By computing the area under the PR-curve, the *Average Precision* (AP) is obtained; averaging AP scores across all classes (species) results in the *mean Average Precision* (mAP). As precision and recall further depend on the degree of overlap a predicted box must achieve with a ground truth label to count as TP, the mAP can be calculated for different IoU thresholds. More precisely, the *mAP50* specifies the mAP at an IoU threshold of 0.5, whereas the *mAP50-95* computes the average mAP that is reached across IoU levels between 0.5 and 0.95. Since it is much harder to maintain high recall scores when an IoU of 0.95 is required for a detection to be considered correct, the mAP50-95 can provide useful insight into the geometric preciseness of a model's predictions.

3.2.2. Point evaluation metrics

For point-based methods, we extend the approach introduced in May et al. (2024) and use the DoR (Eq. (4)) to replace the IoU in the definition of TPs and FPs. I.e., for POLO and HerdNet, we define TPs as detections that match the ground truth class and fall below a certain DoR threshold between the estimated and true location. This allows us to define the *mAP100* and *mAP100-10*, metrics, which measure the mAP reached at DoR thresholds between 1 and 0.1. As is the case for the mAP50/mAP50-95, these metrics can be used to assess model performance under increasingly rigorous demands of spatial accuracy. FPs and FNs are defined as explained in Section 3.2.1.

3.3. Experimental setup

The datasets used in this study are summarized in Table 1, with dataset names specified as composites of the corresponding paper's first author, the location where the images were collected, and the year of publication. For example, *AD-ENCR23* stands for "Alexandre Delplanque", "Ennedi Natural And Cultural Reserve", and 2023. Data collection occurred over the following time periods:

- **AD-ENCR23:** December 2019–January 2020 (Delplanque et al., 2023)

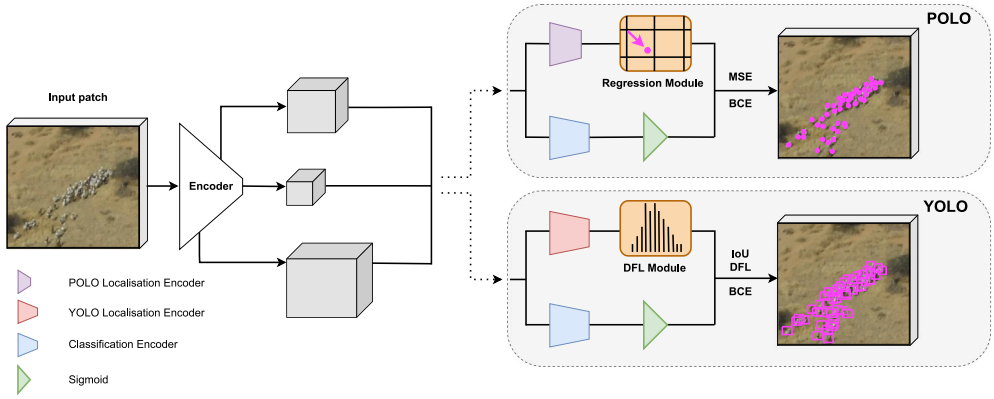


Fig. 1. Diagrams of the POLO and YOLOv8 architecture. In both models the localization and classification encoder consist of a series of convolutions and batch normalization layers, followed by activation functions. The main difference is that the POLO localization encoder uses a reduced amount of channels. Moreover, in YOLOv8, the DFL module serves to sample the probability distribution learned during training to estimate the location of bounding box edges. Its POLO counterpart, the regression module, calculates the coordinates of the animals' center points as described in Eq. (2). Gray cubes represent latent feature maps.

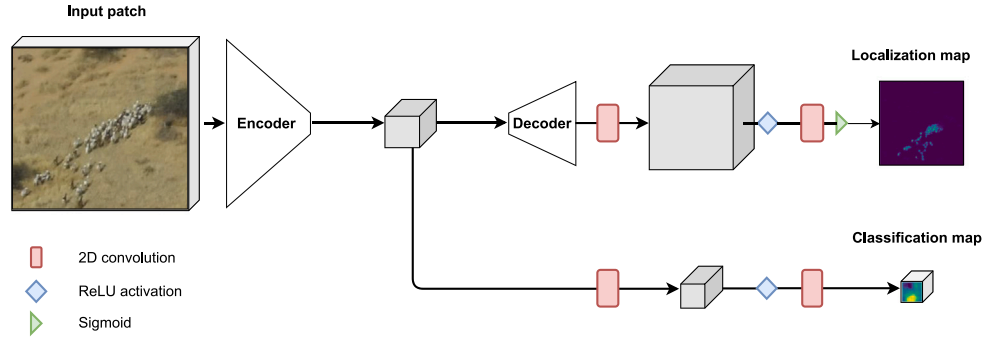


Fig. 2. HerdNet architecture. The resolution of the final localization map is reduced by factor two with respect to the input image, and the classification map is of size 16 × 16 pixels. These output map dimensions have been found to offer the best trade-off between model size and -accuracy during development (Delplanque et al., 2023).

- **BK-L23:** November 2017, March–April 2018 (Koger et al., 2023)
- **EL-IL22:** September–October of 2017-2019 (Weiser et al., 2022)
- **JE-TL19:** March 2014, May 2015 (Eikelboom et al., 2019)

All but the AD-ENCR23 dataset are publicly available. We list the number of annotations per species in the training and test sets in columns N_{train} and N_{test}, and the range and average of the animals' length in pixel in the *Size Range* and *Size Average* columns. For datasets containing hand-crafted bounding box annotations (indicated by the term "boxes" in the *Labels* column), this range is delimited by the minimum and maximum bounding box length found for each species, where we define the length as the larger value between a box's width and height. In cases where only point annotations were available, we obtained size values by manually reviewing a subset of the images. Size variations within a species occur mostly due to the animals' distance to the camera, as well as juveniles will appear smaller than adult specimen. We further use a validation set to monitor model performance during training and to optimize the IoU and DoR threshold for every dataset. The number of validation samples is given in the N_{val} column. Lastly, we calculate the average number of annotations per image for each dataset, which is noted in the *Density* column.

We do not consider thermal datasets for this study as large scale censuses of animal populations are mostly conducted using RGB sensors (Attard et al., 2024; Converse et al., 2024).

3.4. Data processing, model training & inference

All models were trained on overlapping image patches of size 640 × 640 pixels, following best practices for small object detection (Ozge Unel et al., 2019). The amount of overlap in pixels is given in Table 2, and was chosen to ensure that every animal will be fully represented in at least one patch while keeping redundancy between overlapping patches as low as possible. Importantly, the overlap necessary to avoid cutting off animals will depend on the size of the latter, and hence differs between datasets.

When training on either pseudo- or hand-crafted boxes, annotations exceeding patch limits were kept if at least 15% of the box lied within the patch, in order to force the models to detect animals at the image edges. In the case of point labels, annotations cannot be split across patches. Hence, if an animal spans multiple patches but its point label falls outside the current patch, the visible portion within the patch will remain unlabeled. This leads to fewer training annotations compared to the bounding box setting (5%–19% of difference across datasets), where boxes can be clipped to fit patch boundaries.

In addition to patches containing ground truth labels, we also provide the models with image data not showing any animal, which we call *negative samples*. Injecting these negative examples into a model's training helps it distinguish animals from the background more clearly, which (Delplanque et al., 2023) have found to significantly improve

Table 1
Datasets used to benchmark our models.

Name	Species	N_{train}	N_{val}	N_{test}	Size range	Size Avg.	Density	Angle	Labels
AD-ENCR23 (Delplanque et al., 2023)	Camel	2608	380	753	30–130	80	24.95	oblique	points
	Donkey	861	127	239	25–70	50			
	Sheep/Goat	12,486	1774	3579	10–40	25			
BK-L23 (Koger et al., 2023)	Zebra	14,315	3094	3094	12–150	70	20.1	nadir	boxes
	Gazelle	4830	980	980	13–95	50			
	Waterbuck	498	40	40	29–87	60			
	Buffalo	6192	1615	1615	15–140	80			
	Other	854	169	169	18–170	30			
EW-IL22 (Weiser et al., 2022)	Brant Goose	339,801	20,225	64,764	21–180	50	104.54	nadir	points
	Other	35,580	785	4910	21–180	50			
	Gull	4673	374	584	17–150	37.5			
	Canada Goose	38,201	2127	7233	21–180	50			
	Emperor Goose	1746	42	225	21–180	50			
JE-TL19 (Eikelboom et al., 2019)	Elephant	891	140	288	9–204	62	7.67	oblique	boxes
	Giraffe	755	93	261	9–215	81			
	Zebra	1357	219	301	7–126	49			

Table 2
Training hyperparameters for the different datasets. The *Negative Patches* column indicates the amount of empty patches randomly sampled per dataset.

Dataset	Overlap [pixels]	Negative patches [%]	Epochs
AD-ENCR23	128	30	300
BK-L23	192	5	150
EW-IL22	64	5	150
JE-TL19	128	30	300

performance in the context of aerial wildlife detection. We, too, explore the effect of negative samples and find confirmation of their positive impact on detection accuracy. Results are provided in Appendix. Given that there are a lot more negative samples than non-negative patches within our data, we keep only a fixed percentage of the former (see Table 2). We use a considerably larger amount of negative patches when training on the oblique datasets, as this boosted model performance during initial trials with the YOLOv8 and POLO architectures. For the same reasons, we increased the number of epochs when training on these oblique images. Training parameters are summarized in Table 2.

To detect and count animals in the test set, we again divide images into patches, but then map the patch predictions back to the image level to obtain "global" counts and detections. This requires performing an additional round of NMS to remove redundant predictions in image regions where patches overlap. We use the SAHI framework (Akyon et al., 2022) to optimize this process for all bounding box models.

Finally, all YOLOv8_p models are trained on pseudo-boxes and tested on hand-crafted labels. By comparing the geometric accuracy of YOLOv8_p with respect to YOLOv8 in this way, we aim to assess the suitability of pseudo-labels for model training.

4. Experiments & results

In this section, we report the results of our comparisons. We group them by research questions.

4.1. How does label complexity affect model performance?

We assess the effect of label complexity on model performance in both the nadir and oblique scenario by comparing POLO, YOLOv8 trained on pseudo-boxes (YOLOv8_p), and YOLOv8 trained on hand-crafted labels. We train on the BK-L23 and JE-TL19 datasets and use the bounding box centers as point labels for POLO, and as starting points for generating pseudo-boxes. Specifically, we automatically draw squares around the box centers to create the pseudo-boxes, and set the length of all four sides to the size averages specified in Table 1. The same values also serve as radii for training POLO.

Tables 3–5 report the results on the BK-L23 and JE-TL19 test set, after fine-tuning the DoR and IoU threshold on the validation set. Two tasks are assessed: counting (through the MAE score; Table 3), and geometric accuracy (using mAP50-95/mAP100-10; Table 4 and 5). All YOLOv8 models and YOLOv8 derivatives are available in different sizes, where larger models come with a higher number of parameters in the encoder and head of the architecture. For reasons of computational efficiency, we use the smallest model versions (YOLOv8n/YOLOv8n_p/POLO-n).

On both tasks, YOLOv8n achieves the highest counting accuracy. The overall differences between models are relatively small, though, as the discrepancy in MAE never exceeds an absolute value of 0.32 animals per picture, which is found in the *Giraffe* class of the JE-TL19 dataset (1.88 for YOLOv8n vs. 2.2 for POLO-n; cf. Table 3). All models perform a lot worse on the JE-TL19 data compared to BK-L23. Beyond the occlusions and size variations inherent to the oblique setting, this is likely due to the reduced amount of training samples and poor resolution of the JE-TL19 dataset. Specifically, zebras in the BK-L23 data are on average 70 pixels long, which is 8 pixels more than elephants in the JE-TL19 case. In the latter, zebras are only 49 pixels long (cf. Table 1).

It is interesting to observe that YOLOv8n_p and YOLOv8n are much further apart in geometric accuracy than in their MAE scores. We provide additional explanations for this result in section 4.5, but overall, the small amount of discrepancy between the detection models indicates that pseudo-boxes and points yield competitive performance compared to hand-crafted boxes. We further analyze the impact of label-complexity on performance in the below subsection.

4.2. Is performance maintained at equal annotation effort?

Here, we test whether YOLOv8 can maintain superior performance when the annotation effort is reduced to that of training YOLOv8_p and POLO. Based on the assumption that bounding boxes take 7 times longer to create than point labels (Ge et al., 2023), we do so by training additional YOLOv8n models on $\frac{1}{7}$ (randomly selected) of the training data used for the experiments in Section 4.1. Results are shown in Fig. 3.

For both datasets, training data reduction drastically decreases counting performance, pushing YOLOv8n below both point-label baselines, YOLOv8n_p and POLO-n. More specifically, for most classes, the MAE is higher by at least factor two between POLO-n and YOLOv8n $\frac{1}{7}$.

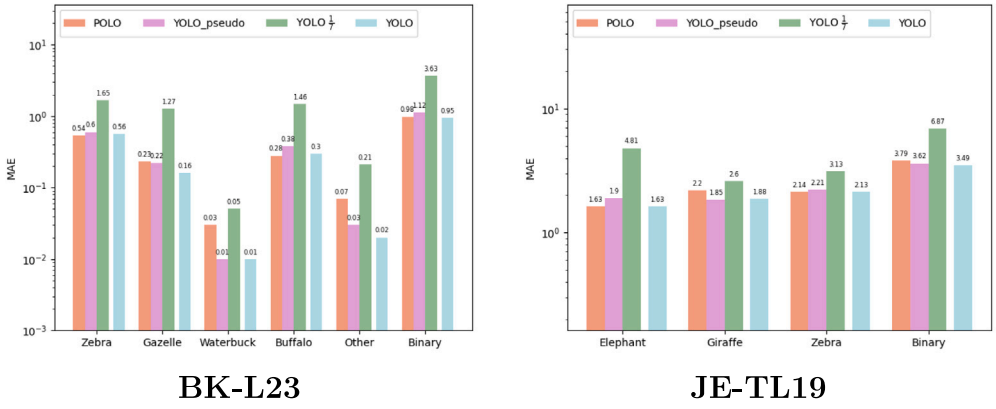


Fig. 3. MAE scores for all YOLOv8-derived models on the BK-L23 and JE-TL19 datasets.

Table 3
MAE scores achieved on the BK-L23 and JE-TL19 test sets.

	BK-L23						JE-TL19			
	Zebra	Gazelle	Waterbuck	Buffalo	Other	Binary	Elephant	Giraffe	Zebra	Binary
YOLOv8n	0.56	**0.16**	**0.01**	0.3	**0.02**	**0.95**	**1.63**	1.88	**2.13**	**3.49**
YOLOv8n_p	0.6	0.22	0.01	0.38	0.03	1.12	1.9	**1.85**	2.21	3.62.
POLO-n	**0.54**	0.23	0.03	**0.28**	0.07	0.98	1.63	2.2	2.14	3.79

Table 4
mAP50-95 scores for YOLOv8n and YOLOv8n_p on the BK-L23 and JE-TL19 test sets.

	BK-L23						JE-TL19			
	Zebra	Gazelle	Waterbuck	Buffalo	Other	Avg.	Elephant	Giraffe	Zebra	Avg.
YOLOv8n	**0.73**	**0.75**	**0.87**	**0.73**	**0.76**	**0.77**	**0.25**	**0.26**	**0.22**	**0.24**
YOLOv8n_p	0.17	0.15	0.44	0.25	0.19	0.24	0.06	0.05	0.07	0.06

Table 5
POLO-n mAP100-10 scores on the BK-L23 and JE-TL19 test sets.

	BK-L23						JE-TL19			
	Zebra	Gazelle	Waterbuck	Buffalo	Other	Avg.	Elephant	Giraffe	Zebra	Avg.
POLO-n	0.96	0.97	0.99	0.95	0.92	0.96	0.51	0.44	0.39	0.45

4.3. How sensitive is bounding box detection to ill-defined pseudo-labels?

Despite being inaccurate geometrically, YOLOv8n_p seems to offer a reasonable alternative to using hand-crafted labels for counting animals in most datasets, and is much preferable over YOLOv8n $\frac{1}{7}$. Importantly, though, the YOLOv8 framework explicitly incentivizes learning the dimensions of objects through the DFL (cf. Section 3.1.1). This raises questions about the sensitivity of YOLOv8_p models with regard to ill-defined label dimensions, i.e. pseudo-boxes that are much smaller/larger than the average animals in the training images. Such inaccuracies can easily occur if the images based on which said dimensions are defined contain only a small part of the animals' size range, or if the altitude at which the images were taken varies considerably in the dataset.

We investigate this matter by increasing/decreasing the size of pseudo-boxes up to factor 2 and down to factor 0.5 in steps of 0.25 from the dimensions specified in Table 1. We then train separate YOLOv8n_p models for each size-increase/reduction. The distribution of MAE scores for the BK-L23 and JE-TL19 dataset is summarized in Fig. 4, detailed numerical results are reported in Tables 6 and 7.

The effect of label inaccuracy is notable, but counting performance is relatively robust: For the BK-L23 dataset, the difference between best and worst MAE score amounts to at most 0.35, whereas for the JE-TL19 dataset, it is 0.51. It can further be observed that for most classes the MAE can be improved from the results obtained with the original pseudo-labels. Counting accuracy, however, does not behave monotonically, and we did not identify a clear relationship between model count

Table 6
MAE scores achieved under increasing label inaccuracy on the BK-L23 dataset.

Factor	Zebra	Gazelle	Waterbuck	Buffalo	Other	Binary
0.5	0.62	0.24	**0.01**	0.4	**0.03**	1.18
0.75	0.59	**0.22**	0.01	0.43	0.03	1.18
1.0	**0.6**	0.22	0.01	0.38	0.03	**1.12**
1.25	0.73	0.25	0.01	0.35	0.07	1.23
1.5	0.72	0.26	0.02	0.37	0.04	1.3
1.75	0.78	0.26	0.01	**0.31**	0.05	1.27
2.0	0.76	0.31	0.02	0.4	0.05	1.47

Table 7
MAE scores achieved under increasing label inaccuracy on the JE-TL19 dataset.

Factor	Elephant	Giraffe	Zebra	Binary
0.5	**1.68**	1.92	2.26	3.75
0.75	1.91	2.31	2.08	**3.57**
1.0	1.9	2.0	2.42	3.61
1.25	1.8	2.41	2.24	3.6
1.5	1.99	2.05	2.21	3.84
1.75	1.99	2.02	2.12	3.6
2.0	1.95	**1.9**	**2.05**	3.63

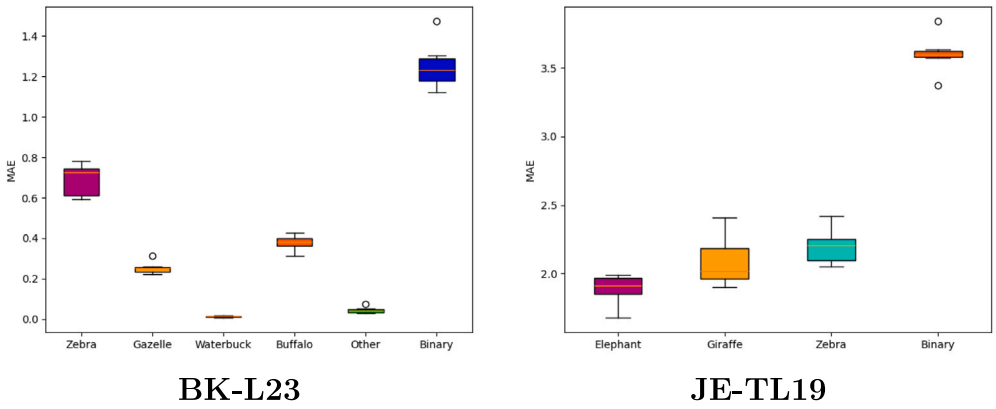


Fig. 4. Distribution of counting accuracy using noisy labels.

Table 8
Binary counting performance for each dataset.

	AD-ENCR23		BK-L23		EW-IL22		JE-TL19	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
YOLOv8n $\frac{1}{7}$	–	–	3.63	85.45	–	–	6.87	137.90
YOLOv8x $\frac{1}{7}$	–	–	2.61	70.26	–	–	7.56	161.31
YOLOv8n_p	8.72	230.66	1.12	5.70	7.42	356.6	3.62	22.62
YOLOv8x_p	9.03	270.36	**0.86**	5.46	7.31	300.79	**3.26**	24.71
POLO-n	9.47	275.23	0.98	**5.44**	6.34	344.0	3.79	27.46
POLO-x	9.51	288.23	1.07	17.23	**4.48**	**173.4**	3.4	26.94
HerdNet	**8.51**	**201.29**	1.56	18.86	8.81	377.24	3.33	**21.0**

error and pseudo-box dimensions within the tested inaccuracy levels. Optimizing these dimensions comes at a high computational cost, and risks to cancel out the reduction in labeling effort gained from using pseudo-boxes. Considering the robustness of model performance, we conclude that such optimization is not necessary and use the pseudo-dimensions originally defined in Table 1 (factor 1.0) for the remainder of this work.

4.4. At the same annotation effort, how does SOTA object detection compare to I2M approaches?

We turn to our final research question and compare YOLOv8_p, POLO, and HerdNet on all datasets. Where possible – i.e., on datasets that come with hand-crafted bounding boxes – we include YOLOv8 models trained on $\frac{1}{7}$ of the data. Moreover, we add results for the largest versions of all YOLOv8 models and YOLOv8 derivatives (YOLOv8x/YOLOv8x_p/POLO-x) to see if increasing the models' size helps improve performance. Table 8 contains the counting results, whereas the geometric evaluation is provided in Tables 9 and 10.

Our results show that, while larger models can help improve model performance, it is not always the case. For example, on the EW-IL22 and JE-TL19 datasets, using larger versions of YOLOv8_p and POLO improves counting and geometric accuracy, whereas in the AD-ENCR23 and BK-L23 case, bigger models improve geometric performance but lower the counting accuracy. This means that, interestingly, models can be worse at counting and yet more accurate geometrically. Similar behavior can be observed between POLO-n and POLO-x on the AD-ENCR images, where POLO-x yields better geometric results and worse counting performance. We further discuss this matter in the next section.

Even if results for JE-TL19 images improve with model size, the dataset remains problematic for YOLOv8-based architectures, and HerdNet shows comparable performances. All architectures give high MAE values on the AD-ENCR23 data, which we attribute to a combination of occlusion, scale variation and low resolution. As it can be seen from Table 1, the size of the most abundant class in this dataset, "Sheep/Goat", amounts to only 25 pixels on average. HerdNet performs best on the AD-ENCR23 data, though the difference is minimal in some cases (8.72 MAE for YOLOv8n_p vs. 8.51 MAE for HerdNet). We also notice that lower MAE scores do not always lead to lower MSE values. In general, large errors are penalized more severely by the MSE (since the metric is squared) compared to MAE, meaning that models with good MAE and worse MSE may on average make less mistakes, but have more significant outliers in their counting performance.

All models are significantly more accurate on the datasets with nadir acquisitions, both in counting and geometrically. While the EW-IL22 data seems more challenging for HerdNet and YOLOv8_p, due to elevated MAE scores, the average density of animals must also be taken into consideration. In this dataset, it is more than ten times higher compared to JE-TL19 (cf. Table 1).

Disregarding YOLOv8 $\frac{1}{7}$, YOLOv8-based models outperform HerdNet in counting on three out of four datasets. In two out of those cases, YOLOv8_p performs better than POLO, the exception being the EW-IL22 data. On the BK-L23 images, YOLOv8x_p counts are more accurate than those of POLO-x, but YOLOv8n_p shows higher MAE than POLO-n, suggesting that the increased number of parameters helps YOLOv8x_p leverage the additional spatial information available in the pseudo-boxes to reduce both, false positives and negatives.

4.5. Additional analyses

We further investigate some of the more noteworthy observations made throughout our experiments. First and foremost, we have found that models with better geometric accuracy can exhibit worse counting performance. For example, looking at the evaluation metrics of POLO-n and POLO-x for the AD-ENCR dataset (cf. Tables 8 and 9), we see that POLO-x yields better mAP, recall, and confusion, but worse MAE values. The improved mAP of POLO-x indicates that the model produces significantly less FPs, which also explains the drastically lower total animal count in Table 11. Intuitively, given this low number of total predictions, one would expect POLO-x to miss a significant amount of animals and therefore score worse in recall with respect to POLO-n, but this behavior is not confirmed. A possible explanation is that higher recall values only indicate a better ratio between TPs and FNs, and not necessarily a lower absolute number of FNs. It is hence entirely possible that POLO-x misses more animals, and at the same time provides a higher recall.

Table 11 also shows that models with similar MAE scores can differ significantly in the accuracy of total counts. This is particularly evident when comparing POLO-n to POLO-x, and YOLOv8n_p to YOLOv8x_p on the AD-ENCR23 and JE-TL19 datasets. To better explain this behavior, Fig. 5 shows the frequency of counting errors for YOLOv8n_p, POLO-n, and POLO-x on the AD-ENCR dataset. From

Table 9
Evaluation metrics for the point models (mAP/recall = Average mAP100-10/recall across classes). Confusion is defined as per Eq. (9).

	AD-ENCR23			BK-L23			EW-IL22			JE-TL19		
	mAP	Recall	Confusion	mAP	Recall	Confusion	mAP	Recall	Confusion	mAP	Recall	Confusion
POLO-n	0.41	0.41	0.21	**0.96**	**0.95**	**0.0**	0.73	0.75	0.14	0.45	0.41	0.12
POLO-x	**0.6**	0.45	**0.08**	0.96	0.95	0.0	**0.75**	**0.77**	**0.13**	0.56	0.45	0.07
HerdNet	0.58	**0.51**	0.12	0.93	0.92	0.0	0.61	0.65	0.19	**0.61**	**0.48**	**0.03**

Table 10
Evaluation metrics for the box models (mAP/recall = Average mAP50-95/recall across classes). Confusion is defined as per Eq. (9).

	AD-ENCR23			BK-L23			EW-IL22			JE-TL19		
	mAP	Recall	Confusion	mAP	Recall	Confusion	mAP	Recall	Confusion	mAP	Recall	Confusion
YOLOv8n $\frac{1}{7}$	–	–	–	0.54	0.74	0.01	–	–	–	0.01	0.04	0.4
YOLOv8x $\frac{1}{7}$	–	–	–	**0.6**	**0.81**	0.03	–	–	–	0.01	0.04	0.44
YOLOv8n_p	0.31	0.4	**0.03**	0.24	0.72	0.01	**0.56**	0.7	0.14	0.06	**0.23**	0.11
YOLOv8x_p	**0.47**	**0.45**	0.08	0.24	0.73	**0.0**	0.56	**0.75**	**0.1**	**0.08**	0.22	**0.07**

Table 11
Total counts obtained with the different models.

	AD-ENCR23	BK-L23	EW-IL22	JE-TL19
YOLOv8n $\frac{1}{7}$	–	5518 (−6.4%)	–	707 (−16.8%)
YOLOv8x $\frac{1}{7}$	–	5475 (−7.2%)	–	1143 (+34.5%)
YOLOv8n_p	**4645 (+1.6%)**	5945 (+0.8% %)	81,075 (+4.3%)	858 (+0.9%)
YOLOv8x_p	3479 (−23.9%)	5809 (−1.5% %)	82,728 (+6.4%)	584 (−31.3%)
POLO-n	4679 (+2.4%)	**5871 (−0.4%)**	81,764 (+5.2%)	**813 (−4.3%)**
POLO-x	3310 (−27.5%)	5689 (−3.5%)	**80,062 (+3.0%)**	531 (−37.5%)
HerdNet	3534 (−22.7.0%)	5665 (−3.9%)	83,730 (+7.7%)	567 (−33.3%)
True Count	4571	5898	77,716	850

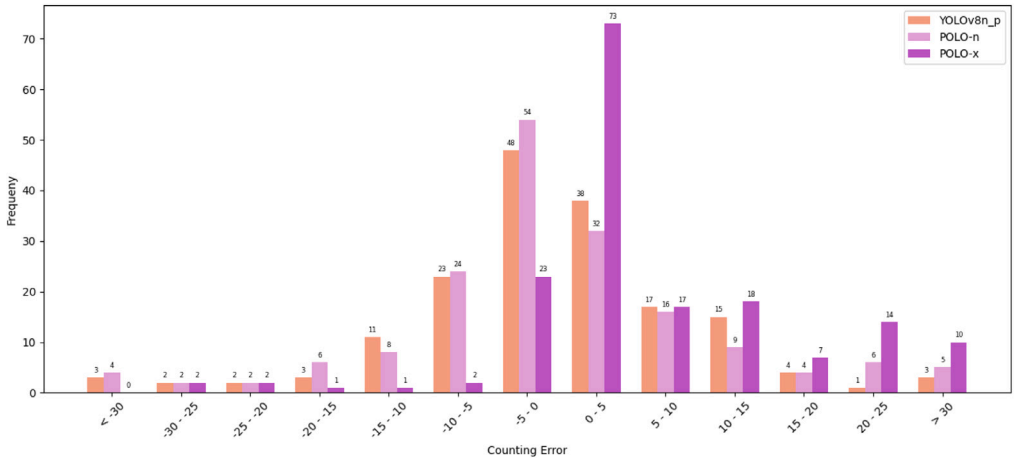


Fig. 5. Error distribution on the AD-ENCR dataset for YOLOv8_p, POLO-n, and POLO-x. Negative error values indicate over-detection, i.e., too many animals being predicted, whereas positive errors mean the opposite. As can be seen, YOLOv8n_p and POLO-n achieve a more balanced ratio between over- and under-detecting compared to POLO-x, which leads to more accurate total counts as positive and negative errors cancel each other out.

these results, we understand that accurate total counts are the result of counting errors being distributed more uniformly across both the positive and negative value ranges. As a consequence, these models also exhibit more balanced mAP and recall scores (see Tables 9 and 10.

Secondly, it was observed that the geometric accuracy of YOLOv8_p is much lower than that of YOLOv8, while counting performances are not as dissimilar (see Tables 3 and 4). When training on pseudo-boxes, models are taught to predict squares of uniform dimensions, which does not capture potential variations in the animals' size and orientation, and will cause low overlap with the ground truth in many cases. We show two examples in Fig. 6. It follows that low mAP values result mainly from this lack of geometric correspondence, and not so much from an increased amount of FPs.

Finally, accuracies differ strongly between HerdNet and either POLO version for the EW-IL22 dataset. We find through qualitative examination that FPs come more commonly in the form of double/triple predictions of the same animal than background detections. See Fig. 7 for an example.

5. Discussion

In this paper, we first assessed the effect of reduced label complexity on SOTA object detection models in the context of aerial wildlife censuses. The results presented in Sections 4.1 and 4.4 bring us to the conclusion that using pseudo-boxes or point labels leads to a notable reduction in annotation time (of factor 7) and an acceptable reduction in counting performance, with only slight differences between these two annotation types. However, training on pseudo-boxes incurs a large drop in geometric accuracy.

The additional information contained in hand-crafted bounding-boxes therefore does not seem to enhance model performance to a degree that justifies the elevated annotation load. Given the same time

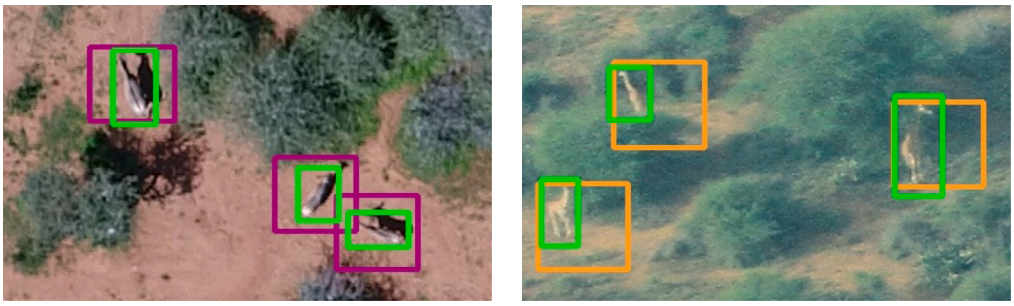


Fig. 6. Visualization of the poorly matching boxes produced by YOLOv8_p models on images from the BK-L23 and JE-Tl19 datasets. Ground truth bounding boxes are highlights in green, whereas model detections are color coded to indicate the predicted animal class (violet = "Zebra", orange = "Giraffe"). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

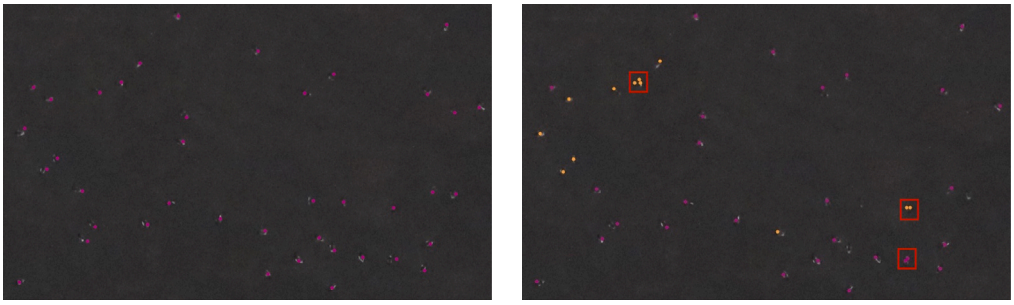


Fig. 7. Detections of POLO-x and HerdNet on an image from the EW-IL22 dataset. Again, colored points correspond to detections, where violet="Brant goose", and orange="other". All pictured birds belong to the "Brant goose" class. Animals for which HerdNet made multiple detections are demarcated by red boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

budget, working with point labels or pseudo-boxes can increase the amount of annotated images up to factor seven, which will yield a much stronger improvement in model performance than higher label complexity. We infer this from the reduced accuracies reported for YOLOv8 $\frac{1}{7}$ in our experiments. Whether or not to generate pseudo-boxes from the point annotations will largely be up to the users' preferences, but considering YOLOv8_p's poor geometric representation of animals, and high counting error on the EW-IL22 dataset (see Table 8), POLO may be the more robust choice.

Our second research question revolves around comparing SOTA object detection models to I2M solutions at equal annotation effort, and under varying data acquisition scenarios. We find advantages of I2M over object detection on oblique datasets containing dense aggregations of small animals, but the gap in performance is much smaller than previously reported (Delplanque et al., 2023). In part, these results can be attributed to optimizing the IoU- and DoR-thresholds. Especially in situations of dense aggregations and partial occlusions of animals, carefully tuning these hyper-parameters helps object detectors to separate close-by individuals. To obtain a general understanding of how the IoU- and DoR-threshold affect model accuracy, we qualitatively assess YOLOv8/POLO predictions at different threshold values on a validation patch of the BK-L23 dataset.

As can be seen in Fig. 8, increasing the DoR is beneficial in the beginning, as it helps suppress redundant detections on the same animal during NMS. But it quickly becomes problematic when detections on neighboring animals start to be suppressed as well, driving up the number of FNs. Thus, for dense aggregations of wildlifes, lower DoR-values are preferable.

With the IoU-threshold, this logic is inverted: to remove multiple detections of the same animal during NMS, the IoU threshold must be lowered, as this will cause predictions to be suppressed even when they overlap only slightly with higher confidence boxes (Fig. 9). Like with the DoR-threshold, this setting is suitable for high animal density situations.

We list the DoR- and IoU-thresholds used throughout this study in Table 12.

Finally, it is important to compare the models used in this study across aspects that go beyond detection capacity and counting performance. For example, conservation practitioners may face situations of limited computational resources where large models will be of no use, regardless of their accuracy. Our I2M model, HerdNet, features ca. 18.7 million parameters, which occupy a maximum of 1.006 Gigabytes (GB) of memory when running inference on a single image patch of size 640 × 640 pixels containing 32-bit floating point values. Inference further requires executing roughly 29.9 billion Floating Point Operations (FLOPs). The small versions of the YOLOv8-based architectures (YOLOv8n/YOLOv8n_p/POLO-n), on the other hand, consist of only 3.2 million parameters and 8.7 billion FLOPs. They require 0.09 GB of GPU memory for single-patch inference. For the large versions (YOLOv8x/YOLOv8x_p/POLO-x), these values are: 68.2 million parameters, 257.8 billion FLOPs, and 0.515 GB of GPU memory at inference time. The reason for the increased GPU usage of HerdNet, despite its lower number of parameters compared to the large YOLOv8-based models, lies in its output: HerdNet returns a localization and classification map (see Fig. 2), which are considerably larger than the boxes/points returned by the YOLOv8/POLO architectures. Especially the small versions of YOLOv8-based architectures will thus be more suitable for deployment in environments of limited compute. This is well supported by the literature, where YOLOv8 has been repeatedly used as a solution for edge computing (e.g., Karim et al., 2024; Bhavana et al., 2024; Ahmed et al., 2023).

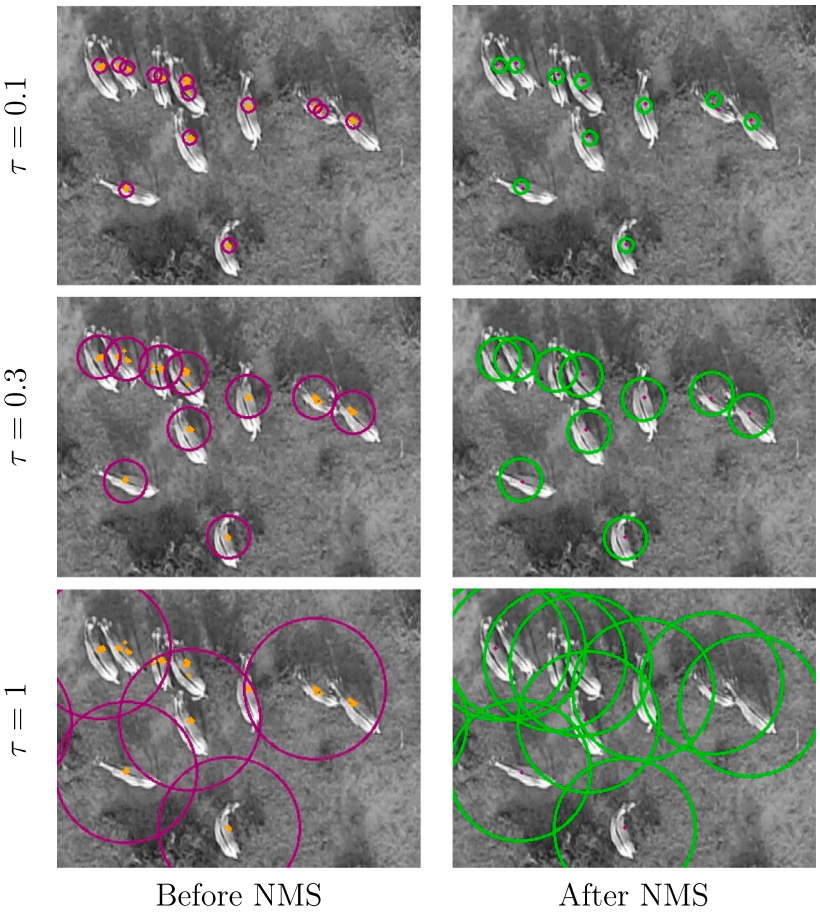


Fig. 8. Effect of the DoR threshold τ on the POLO output. The left column displays predictions (orange dots) before NMS is applied, where purple circles visualize the radius of the highest probability detection that is going to suppress lower probability predictions. The right column shows detections that were retained through NMS (purple dots) along with the area within which estimated locations will be considered TPs (green circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 12
IoU and DoR thresholds used in this study.

	AD-ENCR23	BK-L23	EW-IL22	JE-TL19
IoU	–	0.6	–	0.1
IoU_pseudo	0.9	0.6	0.5	0.3
DoR	0.3	0.3	0.4	0.9

6. Conclusion & future work

We have compared several annotation strategies (bounding boxes, point annotations, and pseudo-boxes) for training animal detection models, and discussed their performance versus ease of deployment and required annotation load. We conclude that in most circumstances state-of-the-art object detection will be a suitable choice for aerial wildlife censuses, and that available modifications to these models significantly improve the balance between counting accuracy and labeling effort. More precisely, a seven-fold reduction in annotation time (Ge et al., 2023) incurs an increase of at most factor 2.5 in MAE (relative difference between YOLOv8n and YOLOv8n_p in the *"Other"* class of the BK-L23 dataset; cf. Table 3). Importantly, this can significantly lower the monetary costs associated with aerial wildlife surveys: After comparing various annotation services, Zhang et al. (2025) find that customers are charged up to three times less for point labels (and thus pseudo boxes) than hand-crafted bounding boxes.

Even at reduced label complexity, object detection models seem able to handle oblique images of animals, if the density is not too high and if animals occupy a sufficient amount of pixels. In the AD-ENCR23 dataset, these conditions are not met (cf. Fig. 10), making an Image-to-Map approach the better choice.

In object detection models, (hand-crafted) bounding boxes can boost performance by injecting size information into the training. However, it has also been found that bounding boxes can introduce ambiguity into the labeling process (Papadopoulos et al., 2017), as it is not always clear how to delineate objects, and the more so in overhead drone images. This not only creates a more challenging annotation task compared to point labels, but it can potentially confuse object detectors. For example, instead of detecting the actual animals, models can learn to latch onto their shadows (Kellenberger et al., 2019), which may not always be present at inference time. Point labels thus offer increased practicality at competitive detection performance. Yet, there are cases in which box labels become indispensable: Tracking algorithms, for example, often use bounding box outputs from a detection model to follow the same animals throughout consecutive frames of a video (Liu et al., 2024). Similarly, bounding boxes can be used for occlusion detection (Saleh and Vámossy, 2022), and for estimating animal biometrics like chest circumference (Zhang and Gu, 2024) - both potentially relevant methods in the context of aerial wildlife monitoring. These applications further require high geometric accuracy and a precise framing of objects, making pseudo-box based solutions unsuitable. Point labeling and the corresponding architectures may therefore suffice for censuses, but more complex use-cases will require hand-crafted bounding box based detection.

Regardless of the architecture and label type, performance on nadir images was considerably higher compared to the oblique datasets,

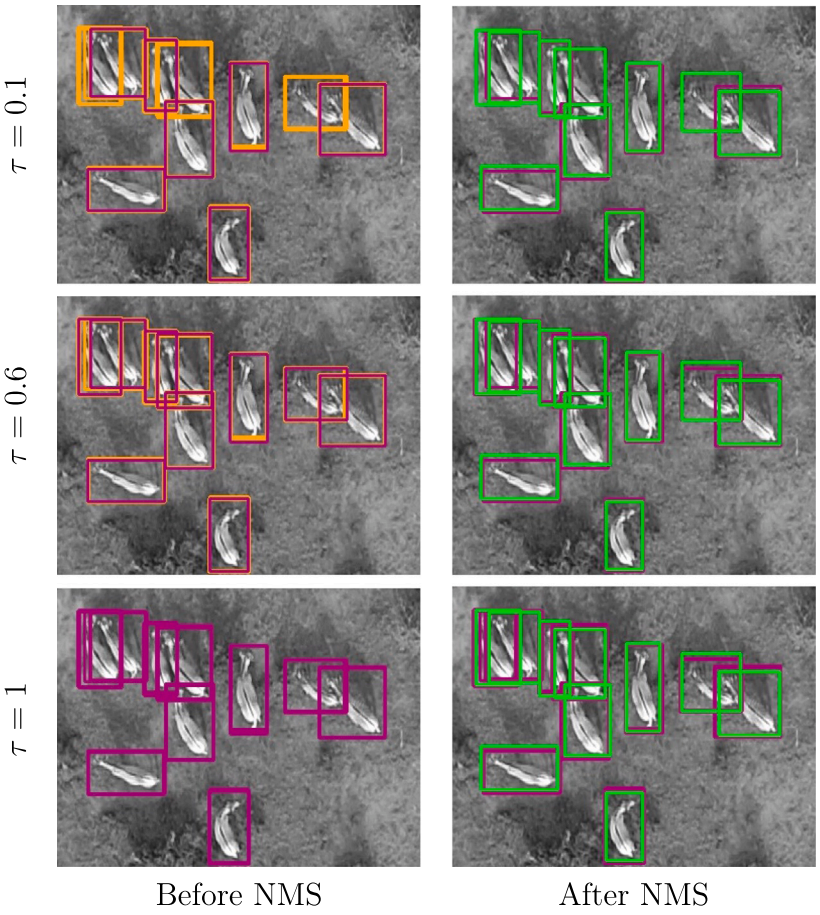


Fig. 9. Effect of the IoU threshold τ on the YOLOv8 output. In the left column orange boxes that share an IoU-value $\geq \tau$ with a purple box will be suppressed. Ground truth boxes are colored green in the right column. Purple boxes sharing an IoU $\geq \tau$ with the latter will be counted as TPs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

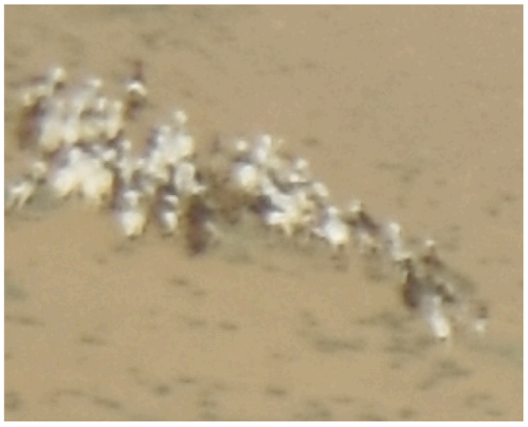


Fig. 10. Example of a dense aggregation typical for the AD-ENCR23 dataset. Animals are between 20–30 pixels in length.

though confounding factors such as low resolution and high diversity of backgrounds limit the conclusiveness of this result. To overcome this limitation, future work should be dedicated to assembling a dataset that allows for a more systematic and isolated evaluation of the camera angle's effect on the performance of different models.

Other factors that can influence the accuracy of object detectors is the velocity of an aircraft, and seasonal changes in the appearance of both, animals and the background. Images taken at high speeds are

Table A.13
Binary counting performance for POLO-n and YOLOv8n_p trained on the AD-ENCR23 dataset with and without negative samples ('w/o neg').

	MAE	MSE
POLO-n	**9.47**	**275.23**
POLO-n w/o neg	11.85	371.29
YOLOv8n_p	**8.72**	**230.66**
YOLOv8n_p w/o neg	13.23	404.68

Table A.14
Evaluation metrics for POLO-n trained on the AD-ENCR23 dataset with and without negative samples ('w/o neg'). mAP/recall = Average mAP100-10/recall across classes).

	mAP	Recall	Confusion
POLO-n	**0.41**	**0.41**	**0.21**
POLO-n w/o neg	0.34	0.39	0.24

Table A.15
Evaluation metrics for YOLOv8n_p trained on the AD-ENCR23 dataset with and without negative samples ('w/o neg'). mAP/recall = Average mAP100-10/recall across classes).

	mAP	Recall	Confusion
YOLOv8n_p	**0.31**	**0.4**	0.03
YOLOv8n_p w/o neg	0.28	0.37	**0.02**

more likely to be blurry, hence creating a more challenging detection task, as well as animals in blurry pictures are more difficult to identify for human annotators, which increases the labeling effort. Certain animals are furthermore known to vary their appearance over the cycle of a year due to shedding or changes in their coat's colors. This can additionally confuse object detection models if the training data does not contain representative samples for all of these variations. The species considered in this work either do not exhibit this behavior or were recorded consistently within the same time period over multiple years (cf. Section 3.3), but this problem may need to be considered in other experimental setups.

Currently, manned aircrafts are used for the majority of large-scale wildlife surveys, and more often than not animals are photographed from an oblique angle during these missions (Delplanque et al., 2023). The rise of drone technology is likely to change this, as well as give conservation practitioners more fine-grained control over data acquisition parameters such as resolution, ground-sampling distance, etc. In combination with an improved understanding of how these parameters influence model behavior – to which we hope to have contributed through this study – this can be expected to increase the robustness and scalability of deep learning based population censuses for wildlife conservation. New problems may emerge as the usage of drones increases among ecologists. For example, it is not trivial to define flight paths that will lead to the right amount of overlap between images, that not only minimizes the risk of double detections and the annotation effort (animals in overlap regions need to be labeled twice), but also guarantees that every animal will appear in the data at least once. These issues are already being addressed (Soares et al., 2024) and represent interesting opportunities for further research.

CRediT authorship contribution statement

Giacomo May: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Emanuele Dalsasso:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Alexandre Delplanque:** Writing – review & editing, Software, Methodology. **Benjamin Kellenberger:** Writing – review & editing, Supervision, Conceptualization. **Devis Tuia:** Writing – review & editing, Supervision, Resources, Conceptualization.

Funding information

This research has been carried out as part of the project WildDrone, funded by the European Union's Horizon Europe Research and Innovation Program under the Marie Skłodowska-Curie Grant Agreement No. 101071224, the EPSRC funded Autonomous Drones for Nature Conservation Missions grant (EP/X029077/1), and the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00280.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Effect of negative samples

We compare `YOLOv8n_p` and `POLO-n` models trained on the AD-ENCR23 dataset with and without negative patches (see Tables A.13–A.15). In all cases, negative patches are crucial for the success of animal detection. The effect of negative sampling on `HerdNet` (and other detection architectures) has been extensively studied, and we refer the reader to Delplanque et al. (2023) for further details.

Data availability

The BK-L23 (Koger et al., 2023) and JE-TL19 (Eikelboom et al., 2019) datasets can be downloaded from the repositories linked in the corresponding publications, whereas the EW-IL22 dataset is available on LILABC. The AD-ENCR23 (Delplanque et al., 2023) dataset cannot be made publicly available as it contains sensitive information. The test images used to obtain the results presented in this paper can be found on Zenodo for all datasets except AD-ENCR23. All code necessary to reproduce our results was uploaded to GitHub.

References

Ahmed, T., Maaz, A., Mahmood, D., ul Abideen, Z., Arshad, U., Ali, R.H., 2023. The yolov8 edge: Harnessing custom datasets for superior real-time detection. In: 2023 18th International Conference on Emerging Technologies. ICET, IEEE, pp. 38–43.

Akyon, F.C., Altinuc, S.O., Temizel, A., 2022. Slicing aided hyper inference and fine-tuning for small object detection. In: 2022 IEEE Int. Conf. Image Process.. ICIP, pp. 966–970. http://dx.doi.org/10.1109/ICIP46576.2022.9897990.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8, 1–74.

Andrew, W., Gao, J., Mullan, S., Campbell, N., Dowsey, A.W., Burghardt, T., 2021. Visual identification of individual holstein-friesian cattle via deep metric learning. Comput. Electron. Agric. 185, 106133.

Arteta, C., Lempitsky, V., Zisserman, A., 2016. Counting in the wild. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part VII 14. Springer, pp. 483–498.

Ascagorta, O., Pollicelli, M.D., Iaconis, F.R., Eder, E., Vázquez-Sano, M., Delrieux, C., 2025. Large-Scale Coastal marine wildlife monitoring with aerial imagery. J. Imaging 11 (4), 94.

Attard, M.R., Phillips, R.A., Bowler, E., Clarke, P.J., Cubaynes, H., Johnston, D.W., Fretwell, P.T., 2024. Review of satellite remote sensing and unoccupied aircraft systems for counting wildlife on land. Remote. Sens. 16 (4), 627.

Bai, H., Mao, J., Chan, S.H.G., 2022. A survey on deep learning-based single image crowd counting: Network design, loss function and supervisory signal. Neurocomputing 508, 1–18.

Bhavana, N., Kodabagi, M.M., Kumar, B.M., Ajay, P., Muthukumaran, N., Ahilan, A., 2024. POT-YOLO: Real-time road potholes detection using edge segmentation based yolo V8 network. IEEE Sens. J..

Bondi, E., Jain, R., Aggrawal, P., Anand, S., Hannaford, R., Kapoor, A., Piavis, J., Shah, S., Joppa, L., Dilkina, B., Tambe, M., 2020. BIRDSAI: A dataset for detection and tracking in aerial thermal infrared videos. In: 2020 IEEE Winter Conference on Applications of Computer Vision. WACV, pp. 1736–1745. http://dx.doi.org/10.1109/WACV45572.2020.9093284.

Borowicz, A., Le, H., Humphries, G., Nehls, G., Höschle, C., Kosarev, V., Lynch, H.J., 2019. Aerial-trained deep learning networks for surveying cetaceans from satellite imagery. PLoS One 14 (10), e0212532.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Ceballos, G., Ehrlich, P.R., Raven, P.H., 2020. Vertebrates on the brink as indicators of biological annihilation and the sixth mass extinction. Proc. Natl. Acad. Sci. 117 (24), 13596–13602.

Chen, A., Jacob, M., Shoshani, G., Charter, M., 2023a. Using computer vision, image analysis and UAVs for the automatic recognition and counting of common cranes (Grus grus). J. Environ. Manag. 328, 116948.

Chen, T., Zhu, L., Ding, C., Cao, R., Zhang, S., Wang, Y., Li, Z., Sun, L., Mao, P., Zang, Y., 2023b. Sam fails to segment anything?–sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, and more. 2, (5), p. 7, arXiv preprint arXiv:2304.09148.

Converse, R.L., Lippitt, C.D., Koneff, M.D., White, T.P., Weinstein, B.G., Gibbons, R., Stewart, D.R., Fleishman, A.B., Butler, M.J., Sesnie, S.E., et al., 2024. Remote sensing and machine learning to improve aerial wildlife population surveys. Front. Conserv. Sci. 5, 1416706.

Delplanque, A., Foucher, S., Lejeune, P., Linchant, J., Théau, J., 2022. Multispecies detection and identification of african mammals in aerial imagery using convolutional neural networks. Remote. Sens. Ecol. Conserv. 8 (2), 166–179.

Delplanque, A., Foucher, S., Théau, J., Bussière, E., Vermeulen, C., Lejeune, P., 2023. From crowd to herd counting: How to precisely detect and count african mammals using aerial imagery and deep learning? ISPRS J. Photogramm. Remote Sens. 197, 167–180.

Delplanque, A., Linchant, J., Vincke, X., Lamprey, R., Théau, J., Vermeulen, C., Foucher, S., Ouattara, A., Kouadio, R., Lejeune, P., 2024. Will artificial intelligence revolutionize aerial surveys? A first large-scale semi-automated survey of african wildlife using oblique imagery and deep learning. Ecol. Inform. 102679.

Desgarnier, L., Mouillot, D., Vigliola, L., Chaumont, M., Mannocci, L., 2022. Putting eagle rays on the map by coupling aerial video-surveys and deep learning. Biol. Cons. 267, 109494.

Doll, O., Loos, A., 2023. Comparison of object detection algorithms for livestock monitoring of sheep in UAV images. In: Int. Workshop Camera Traps, AI, and Ecology.

Dujon, A.M., Ierodiaconou, D., Geeson, J.J., Arnould, J.P., Allan, B.M., Katselidis, K.A., Schofield, G., 2021. Machine learning to detect marine animals in UAV imagery: Effect of morphology, spacing, behaviour and habitat. Remote. Sens. Ecol. Conserv. 7 (3), 341–354.

Duporge, I., Isupova, O., Reece, S., Macdonald, D.W., Wang, T., 2021. Using very-high-resolution satellite imagery and deep learning to detect and count african elephants in heterogeneous landscapes. Remote. Sens. Ecol. Conserv. 7 (3), 369–381.

Eikelboom, J.A., Wind, J., van de Ven, E., Kenana, L.M., Schroder, B., de Knegt, H.J., van Langevelde, F., Prins, H.H., 2019. Improving the precision and accuracy of animal population estimates with aerial image object detection. Methods Ecol. Evol. 10 (11), 1875–1887.

Fabian, Z., Miao, Z., Li, C., Zhang, Y., Liu, Z., Hernández, A., Montes-Rojas, A., Escucha, R., Siabatto, L., Link, A., et al., 2023. Multimodal foundation models for zero-shot animal species recognition in camera trap images. arXiv preprint arXiv:2311.01064.

Gabeff, V., Rußwurm, M., Tuia, D., Mathis, A., 2024. Wildclip: Scene and animal attribute retrieval from camera trap data with domain-adapted vision-language models. Int. J. Comput. Vis. 132 (9), 3770–3786.

Gao, G., Gao, J., Liu, Q., Wang, Q., Wang, Y., 2003. Cnn-based density estimation and crowd counting: A survey. arXiv 2020, arXiv preprint arXiv:2003.12783.

Ge, Y., Zhou, Q., Wang, X., Shen, C., Wang, Z., Li, H., 2023. Point-teaching: weakly semi-supervised object detection with point annotations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, (1), pp. 667–675.

Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.

Gonçalves, B.C., Spitzbart, B., Lynch, H.J., 2020. SealNet: A fully-automated pack-ice seal detection pipeline for sub-meter satellite imagery. Remote Sens. Environ. 239, 111617.

Hoekendijk, J.P., Kellenberger, B., Aarts, G., Brasseur, S., Poiesz, S.S., Tuia, D., 2021. Counting using deep learning regression gives value to ecological surveys. Sci. Rep. 11 (1), 23209.

Hong, S.J., Han, Y., Kim, S.Y., Lee, A.Y., Kim, G., 2019. Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. Sensors 19 (7), 1651.

Jachmann, H., 2012. Estimating Abundance of African Wildlife: An Aid to Adaptive Management. Springer Science & Business Media.

Jackmann, H., 2001. Estimating Abundance of African Wildlife. Kluwer Academic Publishers, London.

Ji, G.P., Fan, D.P., Xu, P., Cheng, M.M., Zhou, B., Van Gool, L., 2023. SAM struggles in concealed scenes–empirical study on segment anything. arXiv preprint arXiv:2304.06022.

Jiang, J., Hu, Y.C., Tyagi, N., Zhang, P., Rimner, A., Deasy, J.O., Veeraraghavan, H., 2019. Cross-modality (CT-MRI) prior augmented deep learning for robust lung tumor segmentation from small MR datasets. Med. Phys. 46 (10), 4392–4404.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO. URL https://github.com/ultralytics/ultralytics.

Jrondi, Z., Moussaid, A., Hadi, M.Y., 2024. Exploring end-to-end object detection with transformers versus YOLOv8 for enhanced citrus fruit detection within trees. Syst. Soft Comput. 6, 200103.

Karim, M.J., Nahiduzzaman, M., Ahsan, M., Haider, J., 2024. Development of an early detection and automatic targeting system for cotton weeds using an improved lightweight YOLOv8 architecture on an edge device. Knowl.-Based Syst. 300, 112204.

Kellenberger, B., Marcos, D., Tuia, D., 2018. Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning. Remote Sens. Environ. 216, 139–153.

Kellenberger, B., Marcos, D., Tuia, D., 2019. When a few clicks make all the difference: Improving weakly-supervised wildlife detection in UAV images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Kellenberger, B., Volpi, M., Tuia, D., 2017. Fast animal detection in UAV images using convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium. IGARSS, IEEE, pp. 866–869.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al., 2023. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026.

Koger, B., Deshpande, A., Kerby, J.T., Graving, J.M., Costelloe, B.R., Couzin, I.D., 2023. Quantifying the movement, behaviour and environmental context of group-living animals using drones and computer vision. J. Anim. Ecol. 92 (7), 1357–1371.

Lalgudi, C.K., Leone, M.E., Clark, J.V., Madrigal-Mora, S., Espinoza, M., 2025. Zero-shot shark tracking and biometrics from aerial imagery. arXiv preprint arXiv:2501.05717.

Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M., 2018. Where are the blobs: Counting by localization with point supervision. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 547–562.

Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images. Adv. Neural Inf. Process. Syst. 23.

Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J., 2020. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. CoRR abs/2006.04388, URL https://arxiv.org/abs/2006.04388, arXiv:2006.04388.

Liang, D., Xu, W., Zhu, Y., Zhou, Y., 2022. Focal inverse distance transform maps for crowd localization. IEEE Trans. Multimed. 25, 6040–6052.

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

Liu, Y., Li, W., Liu, X., Li, Z., Yue, J., 2024. Deep learning in multiple animal tracking: A survey. Comput. Electron. Agric. 224, 109161.

Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H., 2019. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3225–3234.

Lyons, M.B., Brandis, K.J., Murray, N.J., Wilshire, J.H., McCann, J.A., Kingsford, R.T., Callaghan, C.T., 2019. Monitoring large and complex wildlife aggregations with drones. Methods Ecol. Evol. 10 (7), 1024–1035.

Maire, F., Alvarez, L.M., Hodgson, A., 2015. Automating marine mammal detection in aerial images captured during wildlife surveys: a deep learning approach. In: AI 2015: Advances in Artificial Intelligence: 28th Australasian Joint Conference, Canberra, ACT, Australia, November 30–December 4, 2015, Proceedings 28. Springer, pp. 379–385.

May, G., Dalsasso, E., Kellenberger, B., Tuia, D., 2024. POLO–point-based, multi-class animal detection. arXiv preprint arXiv:2410.11741.

Meena, S.D., Manichandana, K.B.V., Potlur, R.S., Dhanyasri, M., Harshith, P., Sheela, J., 2023. Aerial imaging based sea lion count using modified U-net architecture. In: AIP Conference Proceedings, vol. 2869, (1), AIP Publishing.

Michaud, J.S., Coops, N.C., Andrew, M.E., Wulder, M.A., Brown, G.S., Rickbeil, G.J., 2014. Estimating moose (alces alces) occurrence and abundance from remotely derived environmental indicators. Remote Sens. Environ. 152, 190–201.

Morera, A., 2024. Foundation models in shaping the future of ecology. Ecol. Inform. 80, 102545.

Mou, C., Liu, T., Zhu, C., Cui, X., 2023. Waid: A large-scale dataset for wildlife detection with drones. Appl. Sci. 13 (18), 10397.

Naidu, A.P., Gosalia, H., Gakhar, I., Rathore, S.S., Didwania, K., Verma, U., 2025. DEAL-YOLO: Drone-based efficient animal localization using YOLO. arXiv preprint arXiv:2503.04698.

Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., Clune, J., 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc. Natl. Acad. Sci. 115 (25), E5716–E5725.

Ottichilo, W.K., De Leeuw, J., Skidmore, A.K., Prins, H.H., Said, M.Y., 2000. Population trends of large non-migratory wild herbivores and livestock in the Masai Mara ecosystem, Kenya, between 1977 and 1997. Afr. J. Ecol. 38 (3), 202–216.

Ozge Unel, F., Ozkalayci, B.O., Cigla, C., 2019. The power of tiling for small object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Padubidri, C., Kamilaris, A., Karatsiolis, S., Kamminga, J., 2021. Counting sea lions and elephants from aerial photography using deep learning with density maps. Anim. Biotelemetry 9 (1), 27.

Papadopoulos, D.P., Uijlings, J.R., Keller, F., Ferrari, V., 2017. Extreme clicking for efficient object annotation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4930–4939.

Peng, J., Wang, D., Liao, X., Shao, Q., Sun, Z., Yue, H., Ye, H., 2020. Wild animal survey using UAS imagery and deep learning: modified faster R-CNN for kiang detection in tibetan plateau. ISPRS J. Photogramm. Remote Sens. 169, 364–376.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PmLR, pp. 8748–8763.

Rajput, L., Tyagi, N., Tyagi, S., Tyagi, D.K., 2024. State of the art object detection: A comparative study of YOLO and ViT. In: 2024 International Conference on Intelligent Systems for Cybersecurity. ISCS, IEEE, pp. 01–06.

Rančić, K., Blagojević, B., Bezdan, A., Ivošević, B., Tubić, B., Vranešević, M., Pejak, B., Crnojević, V., Marko, O., 2023. Animal detection and counting from UAV images using convolutional neural networks. Drones 7 (3), 179.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Ren, S., Luzi, F., Lahrichi, S., Kassaw, K., Collins, L.M., Bradbury, K., Malof, J.M., 2024. Segment anything, from space? In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 8355–8365.

Rey, N., Volpi, M., Joost, S., Tuia, D., 2017. Detecting animals in african savanna with UAVs and the crowds. Remote Sens. Environ. 200, 341–351.

Ribera, J., Guera, D., Chen, Y., Delp, E.J., 2019. Locating objects without bounding boxes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6479–6489.

Saleh, K., Vámossy, Z., 2022. BBBD: Bounding box based detector for occlusion detection and order recovery. arXiv preprint arXiv:2204.12841.

Schneider, D., Lindner, K., Vogelbacher, M., Bellafkir, H., Farwig, N., Freisleben, B., 2024. Recognition of European mammals and birds in camera trap images using deep neural networks. IET Comput. Vis. 18 (8), 1162–1192.

Seymour, A., Dale, J., Hammill, M., Halpin, P., Johnston, D., 2017. Automated detection and enumeration of marine wildlife using unmanned aircraft systems (UAS) and thermal imagery. Sci. Rep. 7 (1), 45127.

Sharma, N., Scully-Power, P., Blumenstein, M., 2018. Shark detection from aerial imagery using region-based CNN, a study. In: AI 2018: Advances in Artificial Intelligence: 31st Australasian Joint Conference, Wellington, New Zealand, December 11-14, 2018, Proceedings 31. Springer, pp. 224–236.

Soares, V.H.A., Ponti, M.A., Campello, R.J., 2024. Multi-attribute, graph-based approach for duplicate cattle removal and counting in large pasture areas from multiple aerial images. Comput. Electron. Agric. 220, 108828.

Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y., 2021. Rethinking counting and localization in crowds: A purely point-based framework. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3365–3374.

Stapleton, S., Peacock, E., Garshelis, D., 2016. Aerial surveys suggest long-term stability in the seasonally ice-free foxe basin (nunavut) polar bear population. Mar. Mam. Sci. 32 (1), 181–201.

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B.R., Zuffi, S., Risse, B., Mathis, A., Mathis, M.W., Van Langevelde, F., Burghardt, T., et al., 2022. Perspectives in machine learning for wildlife conservation. Nat. Commun. 13 (1), 1–15.

Wang, X., Yan, C., Li, X., Wang, Q., Cui, P., 0000. Comparative evaluation of yolo and rt-detr models for real-time defect detection in wood-based 3d printing, Available at SSRN 5252643.

Weiser, E.L., Flint, P.L., Marks, D.K.S., Brad, S.W., Heather, M.T., Sarah, J.F., Julian, B., 2022. Counts of Birds in Aerial Photos from Fall Waterfowl Surveys, Izembek Lagoon, Alaska, 2017–2019. US Geological Survey, Alaska Science Center.

Wu, Z., Zhang, C., Gu, X., Duporge, I., Hughey, L.F., Stabach, J.A., Skidmore, A.K., Hopcraft, J.G.C., Lee, S.J., Atkinson, P.M., et al., 2023. Deep learning enables satellite-based monitoring of large populations of terrestrial mammals across heterogeneous landscape. Nat. Commun. 14 (1), 3072.

Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., Tomizuka, M., 2022. Autoscale: Learning to scale for crowd counting. Int. J. Comput. Vis. 130 (2), 405–434.

Xu, Z., Wang, T., Skidmore, A.K., Lamprey, R., 2024. A review of deep learning techniques for detecting animals in aerial and satellite images. Int. J. Appl. Earth Obs. Geoinf. 128, 103732.

Ye, Q., Ma, M., Zhao, X., Duan, B., Wang, L., Ma, D., 2025. ADD-YOLO: An algorithm for detecting animals in outdoor environments based on unmanned aerial imagery. Measurement 242, 116019.

Yu, X., Chen, P., Wu, D., Hassan, N., Li, G., Yan, J., Shi, H., Ye, Q., Han, Z., 2022. Object localization under single coarse point supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4868–4877.

Yu, F., Wang, D., Shelhamer, E., Darrell, T., 2018. Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2403–2412.

Zhang, H., Gu, D., 2024. Deep multi-task learning for animal chest circumference estimation from monocular images. Cogn. Comput. 16 (3), 1092–1102.

Zhang, Y., Zhao, S., Gu, H., Mazurowski, M.A., 2025. How to efficiently annotate images for best-performing deep learning-based segmentation models: An empirical study with weak and noisy annotations and segment anything model. J. Imaging Inform. Med. 1–13.

Zhu, P., Peng, T., Du, D., Yu, H., Zhang, L., Hu, Q., 2021. Graph regularized flow attention network for video animal counting from drones. IEEE Trans. Image Process. 30, 5339–5351.