POLO - Point-based, multi-class animal detection

Giacomo May[1], Emanuele Dalsasso[1],
Benjamin Kellenberger[2], and Devis Tuia[1]

[1] EPFL, Switzerland (`first.last@epfl.ch`)
[2] University College London, U.K. (`b.kellenberger@ucl.ac.uk`)

Abstract. Automated wildlife surveys based on drone imagery and object detection technology are a powerful and increasingly popular tool in conservation biology. Most detectors require training images with annotated bounding boxes, which are tedious, expensive, and not always unambiguous to create. To reduce the annotation load associated with this practice, we develop POLO, a multi-class object detection model that can be trained entirely on point labels. POLO is based on simple, yet effective modifications to the YOLOv8 architecture, including alterations to the prediction process, training losses, and post-processing. We test POLO on drone recordings of waterfowl containing up to multiple thousands of individual birds in one image and compare it to a regular YOLOv8. Our experiments show that at the same annotation cost, POLO achieves improved accuracy in counting animals in aerial imagery.

Keywords: Animal detection · Wildlife censuses · Annotation cost

1 Introduction

Frequent animal censuses are a key requirement for successful conservation management and become particularly important when dealing with endangered species. Wildlife in vast and open landscapes can be surveyed efficiently using aerial imagery recorded either from airplanes or unmanned aerial vehicles (UAVs), with the latter being used more and more thanks to the reduced operating costs and safety risks [9,13]. Given the large amounts of data collected during these flights, machine learning approaches are often employed to count animals in the images, allowing biologists to estimate the development of populations. For this purpose, convolutional neural networks (CNNs) are among the most popular techniques [4–6,10].

While CNNs hold the promise of high detection accuracy, this potential is limited by the volume of labeled data available for model training [1]. Since the creation of labeled data implies manual annotation, often to be provided in the form of bounding boxes, accurate counting of animals from aerial imagery comes at a significant annotation cost, which limits the scalability of deep learning-based conservation efforts. To reduce these costs, bounding boxes can be automatically created from point annotations, which can be obtained with higher speed and are thus much cheaper to produce [7,14]. This approach simply consists of generating square boxes around the point annotations and using these pseudo-labels

to train conventional object detection architectures. However, animals in aerial images are regularly of very small size (*i.e.*, a few pixels in length), partially occluded, difficult to separate if standing close together, or distorted due to perspective and motion blur. All of these factors contribute to animal boundaries being difficult to demarcate, hence significantly affecting the quality of automatically generated boxes.

In the present study, we avoid these issues without raising the annotation cost by developing an object detection framework that can be trained entirely on point labels. We approach this task by modifying the YOLOv8 algorithm [8], one of the most advanced and widely used bounding box detection models. The proposed architecture is called POLO (**P**oint-Y**OLO**). We compare two different loss functions to train it, and introduce a new metric to perform non-maximum-suppression (NMS) on point predictions.

We test POLO on a challenging dataset, which involves counting five species of waterfowl in drone images of the Izembek lagoon in Alaska. Moreover, we compare POLO's performance to that of a regular YOLOv8 trained on pseudo-labels, and find that while the latter approach yields high counting accuracy, results improve when using POLO.

2   Related Work

2.1   Point-based object detection and counting

Point labels represent a popular annotation type in the field of crowd counting, where they are used to train density estimation models that take images as input and output a heat map encoding the estimated density of people in every pixel [2, 12, 19] – an approach motivated by the observation that conventional detection algorithms struggle with scale variation (*i.e.*, people in the background appearing smaller than people close to the camera) and partial occlusion of individuals by other individuals.

In object detection on the other hand, the point format has been used as part of mixed labeled sets, where a small set of bounding box annotations is complemented by a larger set of point labels to train detection models to ouptut bounding box predictions [3, 7, 18].

Our solution differs from the above methods, since we output point detections (rather than density maps or bounding boxes) and do not use any bounding box annotations for training. This makes for a simpler learning task than generating bounding box outputs from point labels, and suffices for our purpose of counting animals in drone imagery. The present work is most similar to the efforts of Song *et al.* [16], who also develop a detection model that trains on and outputs point labels. However, their architecture cannot distinguish between different classes, making it unsuitable for animal censuses involving multiple species.

2.2   The YOLO algorithm

To identify objects, YOLOv8 employs a one-stage detection strategy that divides images into a predefined number of grid cells. Each grid cell is then processed on two separate branches, one predicting bounding boxes that enclose objects lying within the cell, and the other predicting the class of said objects. Importantly, YOLOv8 does not directly regress bounding box coordinates, but samples a probability distribution to obtain the most likely number of pixels by which the four bounding box edges are offset from the center of grid cells.

The training objective of YOLOv8 consists of three loss terms: a *binary cross-entropy* loss ($\mathcal{L}_{BCE}$) for the class predictions, an *Intersection-over-Union* loss ($\mathcal{L}_{IoU}$) to learn the geometric prediction of bounding boxes, and a *Distribution Focal Loss* ($\mathcal{L}_{DFL}$) [11] to optimize the probability distribution used for predicting bounding box offsets. The overall loss value is then computed as a weighted sum of these three components, where the class, IoU, and DFL loss are scaled by factors 0.5, 7.5, and 1.5, respectively.

3   Method

To achieve compatibility with point labels, we apply the following modifications to the YOLOv8 architecture:

1. **Output dimensions:** By default, the number of output channels in the final convolutional layer of YOLOv8 is $4 \cdot 16$. These channels encode probabilities of the different bounding box edges being offset by $[0, 1, ..., 15]$ pixels from the center of a grid cell.

   We change the architecture to predict the x- and y- coordinates of the center points of objects that lie within grid cells, using only two channels:

$$\hat{p}_x = \sigma(a_1) \cdot 2 - 0.5 + c_x \tag{1}$$

$$\hat{p}_y = \sigma(a_2) \cdot 2 - 0.5 + c_y \tag{2}$$

   where $\hat{p}_x$ and $\hat{p}_y$ are the predicted coordinates, a_1 and a_2 are the activation values of a grid cell in the first and second output channel, $\sigma(\cdot)$ is the sigmoid function, and c_x and c_y are the coordinates of the grid cell's top left corner. This way, predicted values can range between -0.5 and 1.5, allowing the model to locate objects that do not fit entirely into one cell. We borrow this approach from YOLOv8's predecessor, YOLOv5 [8], where it is used to regress the center point of bounding boxes.

2. **Loss function:** Neither the IoU-, nor the DFL-loss term can be applied to our point model. We consider the following alternatives:

   (a) Average Hausdorff-Distance [15]:

$$\mathcal{L}_{AH}(\hat{P}, P) = \frac{1}{|P|} \sum_{i=1}^{|P|} \text{min}_{\hat{\mathbf{p}} \in \hat{P}} d(\hat{\mathbf{p}}, \mathbf{p}_i) + \frac{1}{|\hat{P}|} \sum_{j=1}^{|\hat{P}|} \text{min}_{\mathbf{p} \in P} d(\hat{\mathbf{p}}_j, \mathbf{p}) \tag{3}$$

where $\hat{P}$ is the set of predicted points, P is the set of ground truth locations, $|\cdot|$ denotes the cardinality, and $d(\cdot, \cdot)$ is the Euclidean distance.

(b) Mean Squared Error (MSE) [16]:

$$\mathcal{L}_{MSE} = \frac{1}{|P|} \sum_{i=1}^{|P|} ||\mathbf{p}_i - \hat{\mathbf{p}}_i||_2^2 \qquad (4)$$

Again, P denotes the set of ground truth locations and $\hat{\mathbf{p}}_i$ is the prediction corresponding to ground truth $\mathbf{p}_i$.

3. **Postprocessing:** To remove redundant detections, we implement a variation of the traditional NMS algorithm. Specifically, we define the *Distance-over-Radius* (DoR) metric, and use it to replace the IoU. Here, the DoR is computed by dividing the distance of a predicted point to its ground truth label by a radius value r_c specified by the user for each object/animal class in the dataset:

$$DoR = \frac{d(\hat{\mathbf{p}}, \mathbf{p})}{r_c} \qquad (5)$$

with $d(\hat{\mathbf{p}}, \mathbf{p})$ denoting the Euclidean distance between a predicted point and the ground truth location. During NMS, low-confidence detections are removed if their DoR to higher-confidence detections falls below a specified threshold.

4   Experiments

4.1   Experimental Setup

Dataset We evaluate and test all models in this study on a publicly available set of drone images [17] from the Izembek lagoon in Alaska hosted on the LILA BC data repository. The dataset includes 9,267 images of waterfowl of size 8688 × 5792 pixels, and a total of 521,270 pseudo-boxes for the classes "Brant goose" (424,790 boxes), "Canada goose" (47,561 boxes), "Gull" (5,631 boxes), "Emperor goose" (2,013 boxes), and "Other" (5,631 boxes). The data was divided into a training (80%), validation (5%), and test set (15%). Images were split into 640×640 patches with 10% overlap, to match the input size expected by YOLOv8 and POLO. 95% of patches that did not contain animals were discarded. The remaining 5% were used as negative samples to reduce false positive detections. Bounding boxes that span multiple patches were clipped to the patch limits if at least 15% of the area of the box lied within the patch in question.

Implementation Details Throughout experiments, we employ a batch size of 32 and set the training duration to 300 epochs, while activating YOLO's early stopping mechanism with a patience value of 50. Counting accuracy was assessed by applying the trained models to our test set. There too, we split images into

640×640 patches with 10% overlap, but map the patch-level predictions back to image-level after inference to obtain a global animal count for the entire image. An additional round of NMS is applied after mapping patch predictions to the image-level in order to remove redundancy in the patch overlap regions.

4.2   Loss functions comparison

In a first step we compare the counting accuracy achieved with POLO models trained on the Hausdorff and MSE loss, respectively. We set a DoR threshold of 0.3 for post-processing during this initial comparison, and use the YOLOv8 loss balancing scheme mentioned in Sec. 2.2; *i.e.*, we assign the Hausdorff-/MSE-loss a weight of 7.5, and scale the classification loss by 0.5. We use a 40 pixel radius for all classes, except for the Gull category, where we set the radius to 30 pixels. These values were obtained by manually measuring the length of animals in the training images, and adding a buffer of 10-15 pixel to account for variations in ground sampling distance between images, and in the appearance of birds. For example, when in flight, animals will occupy more pixels compared to when they are resting on the water surface due to being closer to the drone and their wings being spread. Tab. 1 displays the mean absolute error (MAE) per image obtained when training POLO with the Hausdorff/MSE loss function.

	Brant Goose	Other	Gull	Canada Goose	Emperor Goose
POLO MSE	**5.91**	**4.57**	0.93	2.26	0.25
POLO Hausdorff	6.57	4.64	**0.91**	**2.19**	**0.22**

Table 1: MAE scores obtained with the MSE/Hausdorff loss function.

As can be seen, the Hausdorff model beats the MSE model in three out of five categories. However, for the most abundant species of the dataset (Brant goose), the MSE loss reduces the MAE considerably. We hence decided to use the latter for all subsequent experiments.

4.3   Loss balancing

To optimize the balance between the MSE and classification loss terms, we introduce a hyperparameter $\alpha \in [1, 9]$, based on the value range used for loss scaling in YOLOv8. We then train a total of nine different models using the DoR threshold and radii specified in Sec. 4.2, where the loss is composed as follows:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{MSE} + (10 - \alpha) \cdot \mathcal{L}_{BCE} \qquad (6)$$

Tab. 2 contains the MAE scores for different values of α. We find the model trained with $\alpha = 1$ to perform consistently well across classes: while it is never the most accurate, it yields the second- or third-lowest MAE score in every category. Consequently, we choose the model trained with $\alpha = 1$ for the remaining experiments.

α	Brant Goose	Other	Gull	Canada Goose	Emperor Goose
1	5.3	4.17	0.73	1.89	0.22
2	5.58	4.11	0.86	1.94	0.22
3	5.48	4.2	0.74	2.03	0.2
4	5.51	4.23	0.69	1.81	0.2
5	5.34	4.07	0.86	1.95	0.22
6	5.63	4.31	0.78	2.02	0.21
7	5.25	4.77	0.83	2.05	0.19
8	5.31	4.28	0.81	2.02	0.22
9	5.72	4.62	0.79	1.97	0.23

Table 2: MAE scores for varying loss weights. The first, second, and third best scores are colored in green, blue, and orange, respectively.

4.4   Radius and DoR Threshold

We finally perform a full grid search over combinations of radius values and DoR thresholds. We probe DoR thresholds between [0.1, 1] and multiply the radii defined in Sec. 4.1 by scaling factors in the range of [0.25, 2]. Fig. 1 visualizes the effect different radii and DoR thresholds have on the MAE achieved for the Brant goose class. As can be
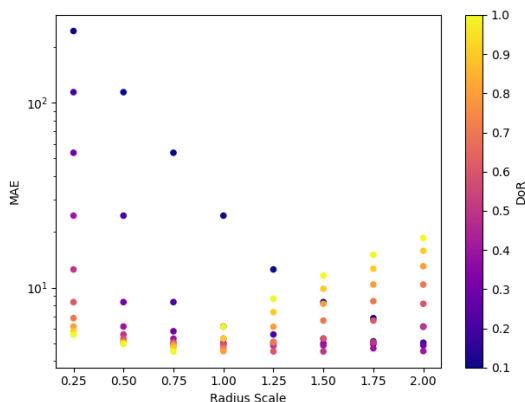

Fig. 1: MAE scores achieved for the Brant goose class depending on radius and DoR value.

seen, two strategies maximize counting accuracy: using intermediate values for both, radius and DoR threshold, or pairing opposite extremes; *i.e.*, combining large radii with low DoR tresholds and vice versa. It should be noted though that the MAE is affected less strongly when large radii are used with high DoR thresholds, compared to the combination of small radii and low DoR values. We observe similar behavior in the remaining classes, where a scaling factor of 1.25 and a DoR-threshold of 0.6 achieve top MAE scores across categories.

5   Results

The animal counts and MAE values obtained with YOLOv8 and POLO are showcased in Tab. 3. Based on the above results, we use a POLO model trained with the MSE loss function scaled by factor 1 in this experiment. We multiply

the radii mentioned in Sec. 4.2 by factor 1.25, and set the DoR threshold to 0.6.

	Brant Goose		Other		Gull		Canada Goose		Emperor Goose	
	Count	MAE	Count	MAE	Count	MAE	Count	MAE	Count	MAE
YOLOv8	68,732	5.86	6,256	**3.78**	1,251	1.05	7,113	1.98	213	0.26
POLO	67,501	**4.51**	6,436	4.08	961	**0.69**	6,923	**1.87**	149	**0.22**
Ground Truth	64,764		4,910		584		7,233		225	

Table 3: Counting accuracy of YOLOv8 and POLO.

Overall, both models yield reasonable counts, but overestimate the abundance of the classes "Brant Goose", "Other", and "Gull" is overestimated , while under-detecting "Canada Goose" and "Emperor Goose". Importantly, POLO achieves a lower MAE in four out of five categories.

Qualitatively, both architectures struggle to separate animals in close proximity of each other, and tend to mistake bright water structures for birds. Differences lie mostly in the classes assigned to these false positive predictions (*cf.* Fig. 2).
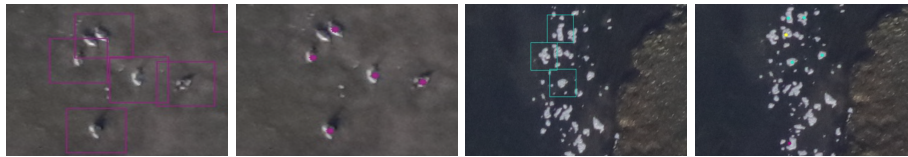

Fig. 2: True- (columns 1 & 2) and false-positive (columns 3 & 4) detections obtained with YOLOv8 and POLO (magenta = Brant Goose, turquoise = Other, yellow = Gull).

6   Conclusion

Our results show that YOLOv8 achieves surprisingly accurate animal counts when trained on pseudo-labels. However, we manage to improve counting accuracy with POLO, and obtain consistently lower MAE scores across categories. As this leads to more conservative estimates compared to YOLOv8, POLO appears less prone to false positive detections. The modifications proposed in this work hence represent an improved way to minimize the annotation cost of conventional object detection. Subsequent efforts will be directed towards assessing the difference between POLO and YOLOv8 when hand-crafted bounding boxes are used for training the latter. We further plan to study the behavior of POLO under different data acquisition scenarios, such as flight altitude, camera angle, etc.

References

1. 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.: Review of deep learning:

concepts, cnn architectures, challenges, applications, future directions. Journal of big Data **8**, 1–74 (2021) 1

2. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 640–644 (2016) 2

3. Chen, L., Yang, T., Zhang, X., Zhang, W., Sun, J.: Points as queries: Weakly semi-supervised object detection by points. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8823–8832 (2021) 2

4. Delplanque, A., Foucher, S., Lejeune, P., Linchant, J., Théau, J.: Multispecies detection and identification of african mammals in aerial imagery using convolutional neural networks. Remote Sensing in Ecology and Conservation **8**(2), 166–179 (2022) 1

5. Dujon, A.M., Ierodiaconou, D., Geeson, J.J., Arnould, J.P., Allan, B.M., Katselidis, K.A., Schofield, G.: Machine learning to detect marine animals in uav imagery: Effect of morphology, spacing, behaviour and habitat. Remote Sensing in Ecology and Conservation **7**(3), 341–354 (2021) 1

6. Eikelboom, J.A., Wind, J., van de Ven, E., Kenana, L.M., Schroder, B., de Knegt, H.J., van Langevelde, F., Prins, H.H.: Improving the precision and accuracy of animal population estimates with aerial image object detection. Methods in Ecology and Evolution **10**(11), 1875–1887 (2019) 1

7. Ge, Y., Zhou, Q., Wang, X., Shen, C., Wang, Z., Li, H.: Point-teaching: weakly semi-supervised object detection with point annotations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 667–675 (2023) 1, 2

8. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics 2, 3

9. Kellenberger, B., Marcos, D., Tuia, D.: Detecting mammals in uav images: Best practices to address a substantially imbalanced dataset with deep learning. Remote sensing of environment **216**, 139–153 (2018) 1

10. Kellenberger, B., Volpi, M., Tuia, D.: Fast animal detection in uav images using convolutional neural networks. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS). pp. 866–869. IEEE (2017) 1

11. Li, X., Lv, C., Wang, W., Li, G., Yang, L., Yang, J.: Generalized focal loss: Towards efficient representation learning for dense object detection. IEEE transactions on pattern analysis and machine intelligence **45**(3), 3139–3153 (2022) 3

12. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1091–1100 (2018) 2

13. Linchant, J., Lisein, J., Semeki, J., Lejeune, P., Vermeulen, C.: Are unmanned aircraft systems (uas s) the future of wildlife monitoring? a review of accomplishments and challenges. Mammal Review **45**(4), 239–252 (2015) 1

14. Mullen Jr, J.F., Tanner, F.R., Sallee, P.A.: Comparing the effects of annotation type on machine learning detection performance. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops. pp. 0–0 (2019) 1

15. Ribera, J., Guera, D., Chen, Y., Delp, E.J.: Locating objects without bounding boxes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6479–6489 (2019) 3

16. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: 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 (2021) 2, 4

17. Weiser, E.L., Flint, P.L., Marks, D.K.S., Brad, S.W., Heather, M.T., Sarah, J.F., Julian, B.: Counts of birds in aerial photos from fall waterfowl surveys, izembek lagoon, alaska, 2017-2019 (2022) 4
18. Zhang, S., Yu, Z., Liu, L., Wang, X., Zhou, A., Chen, K.: Group r-cnn for weakly semi-supervised object detection with points. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9417–9426 (2022) 2
19. Zhao, Z., Li, H., Zhao, R., Wang, X.: Crossing-line crowd counting with two-phase deep neural networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. pp. 712–726. Springer (2016) 2