Using Object Detection Models to Identify and Count Arctic Wildlife

Emily Halina
Department of Computing Science
University of Alberta
ehalina@ualberta.ca

Robert Joseph
Department of Mathematical and Statistical Sciences
University of Alberta
rjoseph1@ualberta.ca

Nicholas Rebstock
Department of Computing Science
University of Alberta
nrebstoc@ualberta.ca

Patrick Wyrod
Department of Mathematical and Statistical Sciences
University of Alberta
pwyrod@ualberta.ca

Sandipan Nath
Department of Computing Science
University of Alberta
snath@ualberta.ca

Abstract

The use of Artificial Intelligence (AI) and algorithmic techniques to identify and classify objects from photographs is a fundamental task in the field of computer vision. Despite advancements in deep learning allowing for the creation of general-purpose models for object detection, these models can prove insufficient for specific novel domains. In this paper, we explore one such novel domain: the localization and classification of caribou in the Canadian Arctic. We perform a survey of several different state-of-the-art object detection approaches, providing analysis of the strengths and weaknesses of each approach. We present novel insight into the challenges of the Arctic as a domain for object detection, along with potential solutions to those challenges.

1 Introduction

Object detection is a fundamental computer vision task involving the detection and classification of objects of interest in a given image. There has been extensive prior work exploring different algorithms and methodologies for solving this task in many ecological domains [5]. One subfield of object detection is wildlife detection, which concerns the detection and classification of wild animals in photos taken by "camera traps." These camera traps are cameras which are placed in a species' environment that are designed to take photos in response to a motion or timer trigger. While there has been prior work in wildlife detection within many environments, the Canadian Arctic is an domain of particular interest and importance [2]. Species of wildlife native to the Arctic have been adversely affected by global warming in recent years, leading to loss of habitat and displacement [5]. The development of wildlife detection models for Arctic wildlife could aid ecologists in tracking and tracing the movements and counts of different species, helping to both preserve those species and gain more insight into their changing habitats.

There are a number of factors which make the task of developing an object detector for Arctic wildlife a novel challenge. Despite the relatively robust history of wildlife detection, there has been little work within similar domains to the Arctic. The vast majority of state-of-the-art (SOTA) wildlife

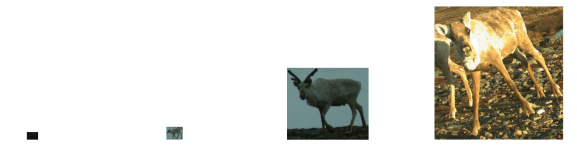


Figure 1: Examples of members of each of our 4 size classes. Note the disparity in detail level of the lower size classes, as well as the comparative clarity of the larger classes.

detection techniques have been developed for environments with dense vegetation where animals may be occluded by branches or trees. In comparison, the Arctic has no such vegetation, causing images to stretch far into the horizon. Due to this, an effective wildlife detection model for the Arctic must be able to able to detect both large and small instances of animals, some of which may only be a few pixels in area. An example of these differences in sizing can be seen in Figure 1.

Toward solving the problem of using object detection to identify and count Arctic wildlife, we investigated and implemented several object detection approaches using a dataset collected in the Arctic across a two year time span. After an initial survey of approaches, we narrowed down to six separate models which we trained to identify caribou from these photos. With input from our domain expert and motivation from prior wildlife detection work, we developed and implemented a novel evaluation metric for judging the success of each approach in our unique domain. From the evaluation of each approach, we gained insight into the nature of the Arctic domain with respect to object detection, as well as potential steps forward for future work.

The contributions of this paper are as follows:

- An overview and implementation of various object detection approaches and techniques with respect to the novel domain of the Canadian Arctic. [1]
- An empirical comparison of six modern object detection approaches on the task of detecting caribou in the Arctic.
- Discussion of our quantitative and qualitative results with respect to the architecture of each approach.

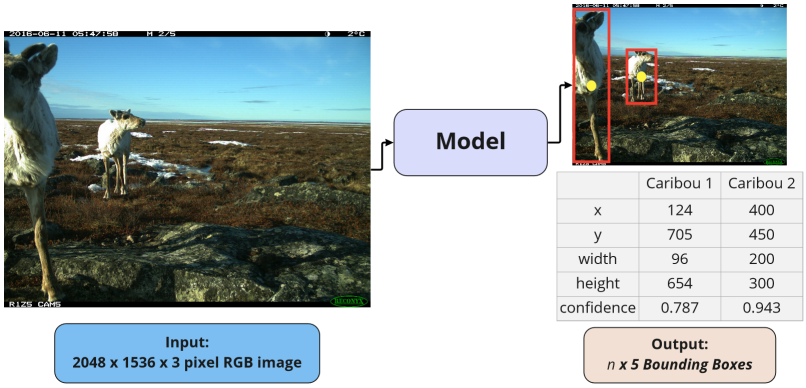
The rest of this paper is as follows. Section 2 provides a formal description of our task, motivating our choice of the detection of caribou and providing a simple example of the system. Section 3 discusses prior work in both general object detection and wildlife detection. Section 4 outlines the different approaches we considered for solving this task, providing reasoning behind why we chose (or did not choose) each. Section 5 gives a formal description of our evaluation metric, motivating it with respect to our task of identifying and counting caribou. Section 6 contains the quantitative results of each of our models along with analysis about each model. Section 7 discusses our qualitative results, along with reflections on our limitations and potential future work.

2 Task Description

Our task is to localize and label all caribou within a given image. Figure 2 shows an example input image alongside our desired output, the locations of all caribou in the image. The task can be formalized as follows: given a single RGB image of size 2,048 by 1,536 pixels, output n predictions in the form $[x, y, w, h, \text{conf}]$. Each prediction represents a bounding box where x and y are the coordinates of the center of the box, w and h are the width and height of the box, and conf is the confidence score of the prediction outputted by the model. As an example, in Figure 2 our input is the image on the left, and the output is the contents of the table. The output is also visualized in the image on the right.

For each labelled image, we have a set of true bounding boxes B which have been created by human annotators which match the form of the model's predictions $\hat{B}$. In order to evaluate these predictions, we start with the highest confidence box in $\hat{B}$, and check if the Jaccard similarity, *i.e.* the Intersection

[1] `https://github.com/Robertboy18/Object-Detection-Arctic-Wildlife`



	Caribou 1	Caribou 2
x	124	400
y	705	450
width	96	200
height	654	300
confidence	0.787	0.943

Figure 2: A simple example of our task, including input and desired output. x and y represent the center of each bounding box, measured in pixels. Width and height represent the width and height of each bounding box, measured in pixels. Confidence represents the model's confidence in the "correctness" of the prediction, which is notably not a calibrated probability.

over Union, between the predicted box and any ground truth box exceeds an chosen threshold (in our case, 0.3). If so, the prediction is a true positive and both the predicted box and corresponding true box are removed from $\hat{B}$ and B respectively. Otherwise, this prediction is a false positive and is removed from $\hat{B}$. This process is repeated until we are left with no more predictions, *i.e.* $\hat{B}$ is empty. At this point, the remaining boxes in B is are considered false negatives.

Our dataset consistent of 1,586 labelled images containing 8,631 caribou in total. These images were taken after the camera trap's motion sensor was triggered. When this occurred, the camera took three images, spaced one second apart. We used a 60%, 20%, 20% split for the training, validation, and test sets respectively. To ensure the test and validation sets did not include images almost identical to those in the training set, we separated all images from each camera, then further separated each camera's images by the day they were taken. Thus, the set of images was partitioned into sets based on both the camera that took them and the date of capture. All images in each of these sets were assigned to one of the training, validation or testing sets. This ensures that any model we train cannot simply memorize nearly identical images.

3 Related Work

There has been prior work on wildlife detection in many domains However, our domain is qualitatively different from the domains studied in other surveys. This is because the vast majority of animals in our images are so far from the camera that they make up roughly 0.03% of the image, while in most domains dense vegetation occludes animals that are not close to the camera. This distribution of sizes can be seen in Figure 3.

In our literature review, we identified the three distinct groups of approaches. The first are one-step approaches, which consider training a single object detection model with classes for each species. The second are two-step approach, which split the task of object detection into two distinct sub-tasks of localizing the location of objects of interest, and then classifying those objects with a different model. The last is background subtraction, which involves the utilization of the temporal structure of image sequences to improve the performance of one of the prior approaches.

One-step approaches have been used to great success for tasks with large amounts of training data. Norouzzadeh *et al.* [11] trained a Convolution Neural Network (CNN)-based model on roughly 1.4 million images from the Snapshot Serenegeti dataset, and were able to achieve human-level performance on some images. Tabak *et al.* [15] were able to achieve 98% accuracy on images from different parts of the United States. However, when testing the same model on an out-of-sample validation set with images from Canada, its accuracy dropped to 82%. This confirms the work of Beery *et al.* [1] in showing that species specific detectors generalize poorly to images containing the same species in different environments. Specifically, Beery *et al.* [1] showed that detection was highly dependant on the texture of the background of the image. Since one-step approaches rely

solely on applying a single object detection model, advances in object detection architectures can quickly be translated to increased performance for one-step approaches.

Norouzzadeh *et al.* [12] showed that it is possible to achieve comparable success to a one-step approach while using fewer samples by employing a two-step approach [11]. In total, it took 14,100 images to match the results, which results in a 99.5% reduction in images labeled. Their detection model MegaDetector (MD) was trained on millions of images from other camera trap datasets, but not on images from the Snapshot Serengeti dataset. They demonstrated that a generic animal detector shows better generalization—both to new environments, as well as to species outside its training set—than a species-specific animal detector. They hypothesized that their success was due to classifier not focusing on background details. This is because the classifier only receives crops of the detected animals instead of the entire image in two-step approaches [12].

Yousif *et al.* [20] achieved 99.58% image level empty vs. non empty classification accuracy on the Snapshot Serengeti dataset which is the current SOTA. We attempted to combine this approach with the other approaches in order to create a robust, accurate model.

4 Methodology

In this section we will discuss the approaches we took, detailing which were developed until completion, and which were not. We will explore our approaches within the framework described earlier, *i.e.* one-step approaches, two-step approaches, and background subtraction. Additional information for each model is provided in Table 2.

4.1 One-Step Approaches

One-step approaches use a single object detection model that take an input image and directly output the predictions with no additional steps. The approaches we considered are divided by subsection below.

4.1.1 Single Shot Detector (SSD)

Single shot detectors (SSD) by Liu *et al.* [10] are approaches that perform a single forward pass over a CNN-based architecture. SSDs are simpler than approaches that need object proposals because they avoid the need for proposal creation and subsequent pixel or feature resampling phases, encapsulating all processing in a single network. One advantage of SSDs is their ability to perform real-time interference on devices with limited CPU resources, such as smartphones. As well, SSDs have a history of use in camera traps and can be trained very quickly. However, SSDs require vast amount of data, which is a problem since our dataset is relatively small. As well, SSDs perform worse on smaller objects in comparison to larger objects, making them unsuitable for our domain. Due to these issues, we decided to drop this model.

4.1.2 DEtection TRansformer (DETR)

DEtection TRansformer (DETR) is an object detector that employs a transformer encoder-decoder on top of a convolutional backbone [3]. DETR is an end to end model which uses a CNN to extract a compact feature representation which is then passed through an encoder-decoder transformer follow by a simple feed-forward network that makes the final detection prediction. DETR is the most recent SOTA solution for end-to-end object detection and has a straightforward implementation and design. However, DETR has a few flaws that made it unsuitable for use in this project. First, even on state of the art hardware, DETR has an incredibly slow convergence time, making retraining for our specific task impractical. As well, DETR performs worse on small objects in comparison to large objects. This is because DETR does not use high resolution CNN layers (feature maps), which means small objects are less likely to be represented well as mentioned by Zhu *et al.* [22]. The model was dropped, although we discuss its potential merits in Section 7.

4.1.3 You only look once (YOLO/YOLOX)

The YOLO family of models are SOTA object detectors that find objects using a single forward propagation across a CNN. YOLOX streamlines preprocessing in comparison to YOLO by eliminating

the need to generate pre-sized bounding boxes or anchor boxes [6]. As well, YOLOX employs advanced data augmentation techniques which are not commonly used by other one-step detectors which help to improve the model's generalizability. YOLOX and YOLOv4 were chosen to represent the YOLO family due to their SOTA performance on standardized object detection datasets. Notably, a major limitation of the YOLO family of models is the inability to recognise and separate small items in images that appear in groups. This is a significant downside in our domain where there are often herds of small, occluded caribou in images. Despite these constraints, the YOLOX-s and YOLOv4 models were kept for further evaluation and exploration.

4.1.4 EfficientDet

EfficientDet was introduced by Tan *et al.* [17] as an approach of improving model efficiency in object detection. We chose EfficientDet due to its efficiency in comparison to other detectors, using 4x-9x less parameters than other SOTA models while still achieving comparable results. To accomplish this, EfficientDet utilizes a weighted bidirectional feature network (BiFPN) and a customised compound scaling algorithm to keep the architecture relatively small. We went with the baseline model EfficientDet-D0 that uses 64 BiFPN layers and a B0 backbone network [16]. This model was kept for further evaluation.

4.2 Two-Step Approaches

We define two-step approaches as those which perform localization—the location of regions of interest (ROIs)—and classification—the assignment labels to each ROI—independently. This independence allows for greater flexibility and modularity in model design. However, such approaches impose a tighter upper bound on theoretical performance, as information acquired by the classifier cannot be utilized by the independent localizer and vice-versa.

4.2.1 Localization

In the context of this task, the objective of the localizer is to isolate ROIs for classification. A naïve localization algorithm (manual sliding window) must either exhaustively search each image (and thus produce many redundancies), or produce an incomplete set of bounding box candidates, thereby compromising performance [21]. Additionally, width and height adds two degrees of freedom, so the computation of a complete set of proposals has quadratic complexity with respect to total pixel count. Therefore, we concluded that a manual sliding window localizer was infeasible for this task.

The advent of deep learning introduced CNNs, neural networks which create a deep feature representation of their inputs. The subsequent regional CNNs (R-CNN) add a region proposal mechanism allowing these models to perform localization using a CNN representation. At its core however, the regional proposal mechanism is closely related to a manual sliding window, and thus also shares its computation complexity issues. The primary benefit of R-CNNs for our localization task is their ability to be trained; a trained localizer generates efficient representations of input images, and therefore ROIs. MegaDetector (MD) is a wildlife detection model utilizing Faster R-CNN which has been trained on millions of camera trap images [12]. Vélez *et al.* [18] found MD to be a strong general-purpose animal detector with respect to both precision and recall. After initial experimentation, we deemed MD's performance and inference speed satisfactory for localization for our task, as it produced exceptional qualitative results. As such, MD is the only localizer considered in our two-step approaches.

4.2.2 Classification

After the localizer generates a set of proposed ROIs, a sub-image based on its ROIs is created for classification. While an ideal localizer would perfectly and exclusively match all true bounding boxes, minimizing false detection is not strictly necessary as long as the classifier correctly identifies all false detections as false.

Applying statistical classification techniques to an image classification task requires a feature mapping of the image, such as a descriptor. A local descriptor is best suited for our two-step approaches, as the feature mapping must occur on sub-images located by MD. One such descriptor is the Histogram of Oriented Gradients (HOG), which extracts image features by computing intensity gradients and counting their occurrences in localized portions [4]. In conjunction with a linear support-vector

machine (SVM), HOG's performance has proven consistent and effective for a wide range of image classification tasks [4], which we verified for our dataset through a meta-comparison with other feasible descriptors and classifiers. One source of inefficiency with the HOG+SVM image classifier is the high dimension of HOG's transformed feature space approaching the sample count available in this task, thereby compromising the effectiveness of SVM [7]. To mitigate this, we considered two dimensionality reduction techniques based on their simplicity: principal component analysis (PCA) and an autoencoder. By training an SVM on lower-dimension descriptor representations and classifying features projected onto the corresponding latent space, the effects of a small sample size relative to the dimension of HOG descriptors can be mitigated. PCA is efficient and effective for this purpose, however we also considered a single-layer autoencoder [13]. Its primary advantage over PCA is extensibility, such as the latent representation optimization SA-BBMO by Shikalgar *et al.* [14], which was demonstrated to improve performance at minimal cost to computation time. Without alteration, we found that the autoencoder yielded similar performance outcomes to PCA when latent dimension and computation time were held constant. Both methods exceeded the performance of SVM on the default descriptor feature space, thus we consider HOG+SVM with autoencoder dimensionality reduction the dominant configuration for our dataset.

Deep learning approaches for object are generally designed as one-step models. However, deep learning can be applied separately to the individual subtask of classification. One advantage of deep learning classifiers is their ability to learn a set of feature descriptors directly from the dataset without human-authoring, such as binning and orientation parameters in HOG. EfficientNet is one such architecture, which we selected for examination based on its strong ImageNet image classification accuracy [16]. EfficientNet was developed using neural architecture search to improve upon past CNN-based image classification models. This is more intricate than HOG descriptors with an SVM classifier. As such, incorporating EfficientNet into a two-step approach was more computationally expensive than its non Deep learning counterparts. As EfficientNet's broad feature extraction strategy is intrinsically different from HOG+SVM, we chose to examine it as our second and final two-step model.

4.3 Background Subtraction

As Yousif *et al.* [20] achieved SOTA results on the Snapshot Serengeti dataset, we attempted to reimplement the system they describe to hopefully increase recall and reduce false positives in our detection. By applying HOG to the entire image, we are essentially embedding the image in a lower dimensional space. Due to the illumination invariance of HOG generated features, this representation of the image is affected less by changes in lighting [4]. We call this representation a feature map, and by subtracting the feature map of another image from the feature map of the current image, we get a representation of where the image changed. The feature maps being more resistant to changes in lighting results in less false detections compared to naïvely subtracting the pixel intensities of the two images.

Qualitatively, we found this method was very successful in detecting movement close to the camera, but did not help with caribou further away. We hypothesize that this is due to the fact that as the objects we are trying to detect approach the size of the HOG cells, it becomes increasingly unlikely that the HOG features will capture information that helps distinguish an animal from the background. Since detecting animals that are very far from the camera was the major challenge we were hoping to solve, and this approach did not help in this endeavor, we decided to drop this approach. However, we believe replacing HOG with a more fine grained feature extractor (*e.g.* a set of CNN generated feature maps) could lead to success, as discussed in Section 7.

5 Evaluation

Our high-level goal is to develop a wildlife detection model that will help ecologists count and identify caribou in camera trap images. As such, we need an evaluation metric that reflects the needs of ecologists using the prospective model. In evaluating these models, we consider two measures of success: precision and recall. Precision–or positive predictive value–refers to the accuracy of a model across all of its predictions. Recall–or sensitivity–refers to the ratio of true objects that have been

Table 1: 0-1 AP scores on various sizes of bounding boxes. Bounding box sizes are measured in pixels, and represent tiny, small, medium, and large images respectively. Highest values in each column are bolded.

	all sizes	0 to 24^2	24^2 to 96^2	96^2 to 384^2	384^2 to 2048^2
YOLOX	0.518	**0.045**	0.137	0.188	**0.761**
YOLOv4	**0.580**	0.014	**0.217**	0.310	0.402
EfficientDet	0.288	0.002	0.025	0.183	0.737
MegaDetector (MD)	0.322	0.001	0.034	0.307	0.476
MD + SVM	0.288	< 0.001	0.016	**0.365**	0.374
MD + EfficientNet	0.199	0.002	0.048	0.177	0.107

discovered across the dataset. Precision and recall are formally defined as

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn} \tag{1}$$

where tp is the number of true positive or "correct" predictions, fp is the number of false positive or "incorrect" predictions, and fn is the number of false negatives, or "missed" objects.

Using precision and recall, we can obtain a "precision-recall" curve by sorting a model's predictions by descending confidence and plotting the values of recall and precision on the x and y axis respectively. Figure 4 contains an example of a precision-recall curve. The standardized evaluation metric for object detection tasks is Average Precision (AP), which is the area under the precision-recall curve. AP is formally defined as

$$\text{AP} = \int_0^1 p(r)dr \tag{2}$$

where $p(r)$ is the precision recall curve. This metric provides a general measure of how accurate an object detection model is across all values of recall.

However, all values of recall are not equally valuable to an ecologist in practice. In consulting with our domain expert, we learned that low values of recall are next to useless to an ecologist. This is because the major draw of a wildlife detection model for ecologists is the ability to locate and accurately count all of the animals in a given image [19]. If a model has high precision but only finds a quarter of the animals in a dataset, it cannot be reliably used for tracking information such as population density of a species in a given area. This is especially true of our domain, where images can have upwards of 100 caribou that need to be accurately detected. As such, we initially aimed to consider both the AP alongside the area under the precision-recall curve from 0.9 to 1, which we refer to as high Average Precision (hAP). hAP is formally defined as

$$\text{hAP} = \int_{0.9}^1 p(r)dr \tag{3}$$

which allows us to examine the AP while considering only the highest possible values of recall. However, we found that non-zero performance on this metric was unattainable by our current models, which we discuss further in Sections 6 and 7.

Due to the high variance in sizing of animals across our dataset, we divide our results across four different bounding box sizes, measured by the area of the true bounding box in pixels. These categories are $(0, 24^2), (24^2, 96^2), (96^2, 384^2), (384^2, 2048^2)$. Examples of objects in each of these size ranges can be found in Figure 1. Similar divisions are standard practice in common datasets such as COCO by Lin *et al.* [9].

6 Results

Table 1 contains the AP score for each model on each category of bounding box size. As a baseline for our two-step approaches, we provide the results of MD on its own treating each region proposal as a predicted caribou. Overall, YOLOv4 was able to achieve the highest AP across all sizes of bounding boxes, with YOLOX trailing closely behind.

Generally, we found that our models were much less effective on smaller objects in comparison to larger objects. We hypothesize this is due to the massive qualitative difference between large and small objects, as can be seen in Figure 1. While approaches such as YOLOX and EfficientDet have achieved SOTA performance on standardized datasets, we found this does not explicitly transfer to success in our domain. Due to the Arctic's lack of vegetation, small objects are much more plentiful than in standardized datasets such as COCO that are used to evaluate general object detection models. This may suggest that these models are not suitable for use in our domain which is densely populated with small objects.

Overall, the two-step MD models did not perform as well as the one-step family of approaches. While MD represents the SOTA in animal localization in camera trap images, we intuit this disparity in performance is due to the novelty of the domain. This was especially true of the MD + EfficientNet model, which seemed to struggle from a lack of training data and the density of small images. Qualitatively, we interpret these results as the EfficientNet classifier overfitting to the smallest objects in the dataset, and thus being unable to perform well on larger ones.

Notably, none of the models were able to achieve 90% recall on the entirety of our dataset, which draws their practical feasibility into question. However, the individual precision-recall curves for each approach on each bounding box size show that this is not an unfeasible goal, especially on larger objects. In particular, every model was able to achieve > 99% recall on the largest size class, while most models struggled to achieve even 10% recall on the smallest. This may suggest that a single generalized model for every size of object may not be feasible for this task. Figures depicting the precision-recall curves of each model on each size of object are provided in Appendix A.

7 Lessons Learned

Overall, we learned the Canadian Arctic is a more unique domain for wildlife detection then we originally anticipated. The lack of vegetation led to images stretching far into the horizon, allowing for tiny caribou to be present throughout the training set. We learned that this is a non-standard problem which is not covered in the pretraining of major object detection models, and that a specialized approach is likely needed to solve the task. With this in mind, we discuss some of the fundamental limitations of our approaches, as well as potential avenues for future work below.

7.1 Limitations

One of the major challenges of this project was the size of the dataset. While many of the SOTA object detection models are trained on millions of images, we only had 1,586 annotated samples to work with. Due to this limitation, we had to rely heavily on pre-trained models which were not specifically adapted to our domain. One potential solution could be data augmentation, the use of algorithmic techniques to modify or generate new training data. However, after some brief investigation we realized data augmentation was out of scope for this project, as it represents an entirely different problem from the task we present here. Similarly, we were also limited by time and computational resources. As an example, we were forced to abandon the transformer-based model DETR due to the model's incredibly slow convergence time even on high-end hardware. Despite this, we feel that we prioritized the most feasible models given the information we had at the beginning of the term, and learned a lot despite these limitations.

7.2 Future Work

There are many possibilities for future work for wildlife detection in the domain of the Arctic. While our chosen models were unable to accurately detect and classify the smallest objects in the dataset, approaches centered around specifically tiny images may be useful in solving this task. In particular, there are many classifiers which have shown excellent results on datasets of tiny images such as CIFAR-10 which could be considered [8]. We hypothesize that it may be beneficial to consider an approach that uses multiple models for different sizes of images. This is due to the evident qualitative differences between the different size classes. Another future approach could entail the use of the additional context associated with sequences of photos. By using background subtraction techniques discussed in Section 4.3, it could be possible to train a model that performed better on smaller objects by using the context clues the additional photos provide.

Acknowledgements

We want to extend thanks to our domain expert Alastair Franke who consistently went above and beyond to help our team with the project over the semester. We also want to thank Professor Russ Greiner and Spencer von der Ohe for their support and patient instruction throughout the course.

References

[1] S. Beery, G. Van Horn, and P. Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[2] S. Beery, D. Morris, and S. Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

[5] S. C. Davidson, G. Bohrer, E. Gurarie, S. LaPoint, P. J. Mahoney, N. T. Boelman, J. U. Eitel, L. R. Prugh, L. A. Vierling, J. Jennewein, et al. Ecological insights from three decades of animal movement tracking across a changing arctic. *Science*, 370(6517):712–715, 2020.

[6] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[7] D. Hsu, V. Muthukumar, and J. Xu. On the proliferation of support vectors in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 91–99. PMLR, 2021.

[8] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[12] M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic, and J. Clune. A deep active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, 12(1):150–161, 2021.

[13] C. Ricciardi, A. S. Valente, K. Edmund, V. Cantoni, R. Green, A. Fiorillo, I. Picone, S. Santini, and M. Cesarelli. Linear discriminant analysis and principal component analysis to predict coronary artery disease. *Health Informatics Journal*, 26(3):2181–2192, 2020. doi: 10.1177/1460458219899210. URL `https://doi.org/10.1177/1460458219899210`. PMID: 31969043.

[14] A. Shikalgar and S. Sonavane. Optimized auto encoder on high dimensional big data reduction: an analytical approach. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14):526–537, 2021.

[15] M. A. Tabak, M. S. Norouzzadeh, D. W. Wolfson, S. J. Sweeney, K. C. VerCauteren, N. P. Snow, J. M. Halseth, P. A. Di Salvo, J. S. Lewis, M. D. White, et al. Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, 10(4):585–590, 2019.

[16] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[17] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[18] J. Vélez, P. Castiblanco-Camacho, M. Tabak, C. Chalmers, P. Fergus, and J. Fieberg. Choosing an appropriate platform and workflow for processing camera trap data using artificial intelligence. 02 2022.

[19] S. A. Wich and A. K. Piel. *Conservation technology*. Oxford University Press, 2021.

[20] H. Yousif, J. Yuan, R. Kays, and Z. He. Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images. *Ecology and evolution*, 9(4):1578–1589, 2019.

[21] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[22] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Appendix

A Additional Figures & Examples

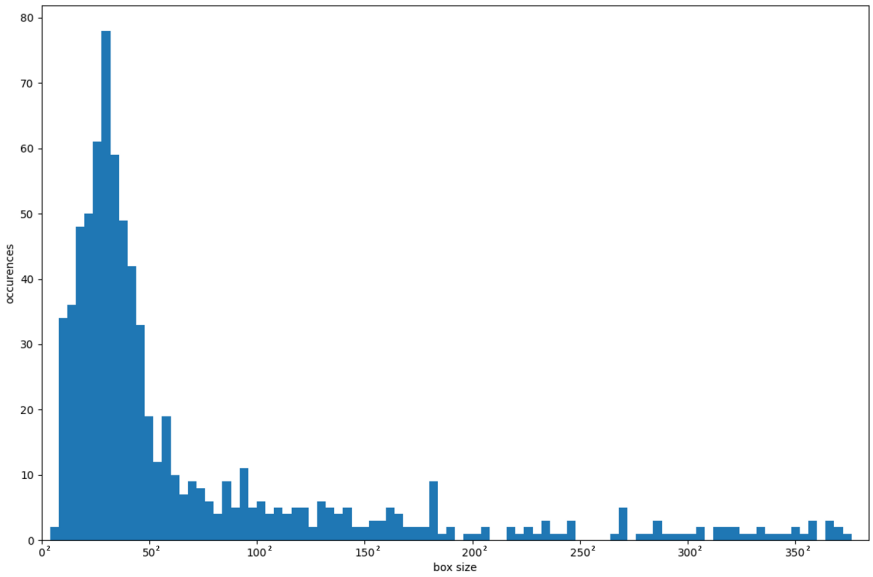


Figure 3: Histogram displaying the distribution of bounding box sizes in terms of the number of pixels. Note, the mode is 32^2 pixels. The histogram omits boxes greater than 384^2 pixels, however in general there exist increasingly few boxes as we increase size.

Table 2: Model Information

Model	Input Size (Pixels)	Parameters
MobileNetV2-SSD	300x300	15M
DETR	480x800	41M
YOLOv4	418x418	60M
YOLOX	640x640	9M
EfficientDet-D0	512x512	4M

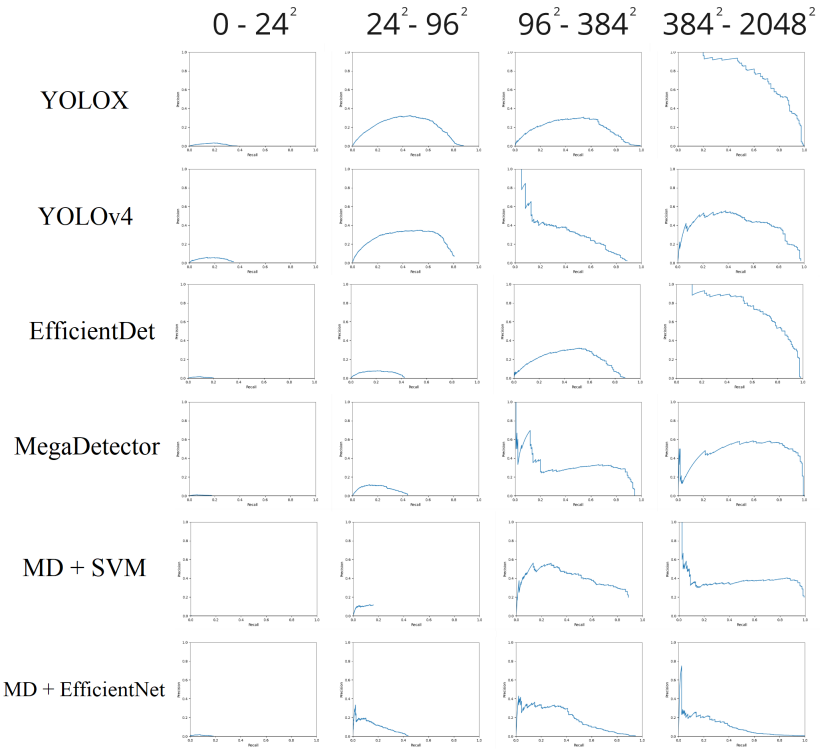


Figure 4: Precision-recall curve for each of our final models on each size class of objects. Precision is on the y axis, and recall is on the x axis. Values are between 0 and 1.

Figure 5: Input image for Figure 6 with 4 true positives.

(a) YOLOv4

(b) YOLOX-s

(c) EfficientDet

(d) MD+EfficientDet

(e) MD+SVM

(f) MD

Figure 6: Predictions of each model on a sample image where each red box denotes a model's prediction and each green box denotes the ground truth.