Deep Learning Object Detection Methods for Ecological Camera Trap Data

Stefan Schneider*, Graham W. Taylor†, Stefan C. Kremer*
*School of Computer Science, University of Guelph
{sschne01, skremer}@uoguelph.ca
†School of Engineering, University of Guelph
gwtaylor@uoguelph.ca
†Vector Institute for Artificial Intelligence
†Canadian Institute for Advanced Research

***Abstract*—Deep learning methods for computer vision tasks show promise for automating the data analysis of camera trap images. Ecological camera traps are a common approach for monitoring an ecosystem's animal population, as they provide continual insight into an environment without being intrusive. However, the analysis of camera trap images is expensive, labour intensive, and time consuming. Recent advances in the field of deep learning for object detection show promise towards automating the analysis of camera trap images. Here, we demonstrate their capabilities by training and comparing two deep learning object detection classifiers, Faster R-CNN and YOLO v2.0, to identify, quantify, and localize animal species within camera trap images using the Reconyx Camera Trap and the self-labeled Gold Standard Snapshot Serengeti data sets. When trained on large labeled datasets, object recognition methods have shown success. We demonstrate their use, in the context of realistically sized ecological data sets, by testing if object detection methods are applicable for ecological research scenarios when utilizing transfer learning. Faster R-CNN outperformed YOLO v2.0 with average accuracies of 93.0% and 76.7% on the two data sets, respectively. Our findings show promising steps towards the automation of the labourious task of labeling camera trap images, which can be used to improve our understanding of the population dynamics of ecosystems across the planet.**

I. Introduction

Population ecologists use camera traps to monitor animal population sizes and manage ecosystems around the world. Camera traps were first introduced in 1956, and in 1995, Karanth demonstrated their usefulness for population ecology by re-identifying tigers (*Panthera tigris*) in Nagarahole, India using a formal mark and recapture model [1], [2]. The popularity of the camera trap methodology grew rapidly thereafter, with a 50% annual growth using the technique as a tool to estimate population sizes [3], [4]. Camera traps respond to motion, which generally corresponds with an animal entering the frame. Camera trap data analyses involve manually quantifying the species and number of individuals in thousands of images. Automating this process has obvious advantages, including a reduction in human labour, an unbiased estimate across analyses, and the availability of species identification without domain expertise.

In this work, we focus on utilizing deep learning based approaches for object detection to identify, quantify, and localize animal species within camera trap images. Camera trap data provides a robust measure of the capabilities of deep learning for species classification, as the images are often 'messy', with animals being partly obstructed, positioned at varying distances, cropped out of the image, or extremely close to the camera [5]. These obstacles are in addition to the traditional difficulties of computer vision tasks, such as variable lighting, photos taken at day and night, and species exhibiting a variety of poses.

Deep learning methods have demonstrated near perfect accuracy for computer vision tasks when trained on large labled datasets; however, labeled ecological data is notorious for being sparse and intermittent [6]. We aim to test the bounds of deep learning for realistic ecological applications, demonstrating the usefulness of the technique for researchers to train their own classifiers on their own ecosystem of interest, instead of relying on large public data sets which may not fit their niche of study. We considered the Reconyx Camera Trap data set, which contains 946 labeled images with 20 species classifications and bounding box coordinates, as well as the Gold Standard Snapshot Serengeti data set, which contains 4,096 labeled images of 48 species classifications [5], [7]. Current methods for object detection require the bounding box coordinates for training, and as a result, we hand-labeled the bounding box coordinates for the Gold Standard Snapshot Serengeti data set and offer it to the camera trap and deep learning community.

We compare two methods for object detection using deep learning, Faster Region-Convolutional Neural Network and You-Only-Look-Once v2.0 (hereafter referred to as Faster R-CNN and YOLO, respectively) [8], [9]. These two approaches are generally considered by the trade-off of data efficiency versus speed, as YOLO can be used in real time, but requires additional training data [8]. Our results demonstrate Faster R-CNN shows promise for accurate and autonomous analysis of camera trap data, while YOLO fails to perform. These results demonstrate that ecologists should consider utilizing Faster R-CNN or its successors as the

method of object detection to autonomously extract ecological information from camera trap images.

II. Background and Related Work

Deep Learning for Object Detection: Many recent advancements in deep learning have come from improving the architectures of a neural network. One such architecture is the Convolutional Neural Network (CNN), which is now the most commonly used architecture for computer vision tasks [10], [11]. CNNs introduce convolutional layers within a network which, for a given image, learn many feature maps which represent the spatial similarity of patterns found within the image (such as colour clusters, or the presence or absence of lines) [12]. Each feature map is governed by a set of 'filter banks', which are matrices of scalar values that can generally be considered synonymous to the weights of a feedforward network. For each convolutional layer, the filter banks are similarly passed through a non-linear transformation and learned using gradient descent with backpropagation [12]. CNNs also introduce max pooling layers, a method that reduces computation and increases robustness by evenly dividing the feature map into regions and returning only the highest activation values [12]. As a result of having numerous feature maps for a given input, CNNs are particularly well suited for dealing with data from multiple arrays, such as colour images, which have three colour channels [12]. Deep learning researchers continually experiment with the modular architectures of neural networks and four CNN frameworks have been standardized as well-performing with differences including computation cost and memory in comparison to accuracy. These networks include AlexNet, VGG, GoogLeNet/InceptionNet (which introduced the inception module), and ResNet, which introduced skip connections [11], [13]–[15]. These networks range from 7 to 152 layers. A common approach to training deep learning classification tasks is to use publicly available weights from one of these four network structures trained on a public data set as initialization parameters, and retraining the network using your own limited data set [16]. This allows for learned filters, such as edge or colour detectors, to be used without having to be re-learned on limited data. This technique is known as Transfer Learning [16].

CNNs have demonstrated great success for image classification, conditioned on the network being trained to return a single label for a given image [11]. In order to determine the classification of more than one object within an image, computer vision researchers train an object detector, where the image is segregated into overlapping regions (often called 'proposals') [17]. Two approaches for object detection have seen wide-spread success. The earliest approach was R-CNN, where an image is crudely segregate into a series of different sized boxes using an image segregation algorithm, and each region is passed through a CNN. Fast R-CNN introduced region proposals generated based on the refined last feature map of the network to decrease proposal computation [17]. Soon after, Faster R-CNN, which introduces a Region Proposal Network (RPN) to the framework, enabled nearly cost-free region proposals [15]. A second approach for object detection is YOLO, which divides an image into a grid, with each gridcell acting as the origin for numerous predefined 'anchors' relevant to the size classifications of interest. For example, when searching for a cat, one may implement three anchors: a square, a horizontal rectangle, and a vertical rectangle, as a cat may approximately fit into each shape. When training and using YOLO, output classifications are returned for every anchor in a single iteration. [8]. YOLO is often less accurate due to the static nature of the anchor boxes, but has been shown to be 3x faster than Faster R-CNN [8].

Automating the Analysis of Camera Trap Images: Prior to the wide-spread adoption of deep learning systems, computer vision researchers developed a variety of creative and moderately successful methodologies for the automated analysis of animals from camera traps based on the raw pixel data from images. Initial approaches for species classification required a domain expert to identify meaningful features for the desired classification (such as the unique characteristics of animal species), design an algorithm to extract these features from the image, and compare individual differences using a statistical analysis. Computer vision systems were first introduced for species classification within the microbial and zooplankton community to help standardized species classification, and considered morphological silhouettes [18]–[20]. The first complete camera trap analysis was done in 2013 using the Scale-Invariant Feature Transformation algorithm in combination with a Support Vector Machine to classify species using the Reconyx Camera Trap data set after a foreground extraction technique was applied to separate the animal from the background [7], [21], [22].

In 2014, Chen et al. [23] reported the first paper for animal species classification using a CNN that considered the Reconyx Camera Trap data set. Their CNN was a shallow network by modern standards, with 3 convolution and 3 pooling layers.

In 2016, Gomez et al. [24] used deep CNNs for camera trap species recognition, comparing 8 variations of the established CNN frameworks AlexNet, VGG, GoogLeNet, and ResNet to train species classification on the complete Snapshot Serengeti data set of 3.2 million images with 48 species classifications. The ResNet-101 architecture achieved the best performance. Following this work, they also utilized deep learning to improve low resolution animal species recognition by training deep CNNs on poor quality images. The data was labeled by experts into two data sets, the first classifying between birds and mammals and the second classification of different mammal species [25], [26].

In 2017, Norouzzadeh et al. [5], utilized the ability of

a network to return numerous output classifications for a given image, a technique known as multitask learning, to consider the species, quantify the number of animals, as well as to determine additional attributes. This approach operates differently than object detection methods, as their classifier learns what an image with a given number of animals looks like, rather than individually detecting the number of individuals within the image. Nine independent architectures were trained, including AlexNet, VGG, GoogLeNet, and numerous variations of ResNet. The authors report a species classification accuracy, counting, and attribute accuracy considering an ensemble of their nine models [5].

These approaches all share the common limitation of returning only one output per classification task per image, which is unrealistic for meaningful camera trap data analyses. Object detection methods account for this limitation, allowing for a classifier to return multiple species as output.

III. Experiments and Results

Reconyx Camera Trap data set and Snapshot Serengeti Project: The Reconyx Camera Trap (further referred to as RCT) data set is a collection of 7,193 camera trap images from two locations in Panama and the Netherlands, capturing colour images during the day, and gray-scale at night [7]. Of all the images, only a subset of 946 images include labeled bounding box coordinates, and so we only considered these images.

The Snapshot Serengeti data set is the world's largest publicly available collection of camera trap images, with approximately 1.2 million images collected using 225 camera traps since 2011 [27]. To provide labels, the organization has created a website where nearly 70,000 individuals help label the images by selecting predefined classifications of the species, the number of individuals (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11-50, 50+), various behaviours (i.e., standing, resting, moving, eating, or interacting), and the presence of young. In additional, there is the Gold Standard Snapshot Serengeti (further referred to as GSSS) data set which contains 4,432 images labeled by experts within the field; however only classification and not bounding box co-ordinates. We annotated the GSSS data set to test object detection methods and give these to the Snapshot Serengeti community. The labeled data set is available at: *https://dataverse.scholarsportal.info/dataset.xhtml?persistentId=doi:10.5683/SP/TPB5ID*.

For this experiment, we consider the ResNet-101 architecture for both object detection methods. ResNet-101 is a robust network that showed great success in other camera trap studies [24]. We initialized both object detection classifiers using a pre-trained model of the Common Object in Context 2017 data set [28]. The weights of the final layer were initialized using the Xavier initialization [29]. Each model was trained using the adaptive momentum optimizer, and training concluded after the loss failed to improve after 3 successive epochs [30].

For both data sets, the bounding box coordinates only pertain to a subset of the larger data set. As a result, there has been no prior experimentation and division of standardized train/test labels. In order to account for this, we perform a cross-validation-based evaluation, repeating the procedure five times and reporting the mean and standard deviation across these runs considering a 80/20 train/test split. To improve accuracy, bounding boxes containing less than 750 pixels were removed from the data set.

We consider two performance metrics, accuracy and Intersection Over Union (IOU). Accuracy represents the percentage of correctly classified species. IOU is an evaluation metric specific to the performance of object detection methods. IOU returns performance as the area of overlap of the true and predicted regions divided by the entire area of the true and predicted regions [31]. To quantify accuracy using object detection, numerous classification comparisons are calculated per image. To do this, we calculate the IOU for each predicted box for an image in comparison to a test box, select the highest IOU, and then compare its classification output to the true classification. After bounding boxes are used for a classification, they are removed for future comparisons. IOU values above 0.70 are considered well performing [31].

Faster R-CNN returned an accuracy of 93.0% and 76.7%, and IOU values of 0.804 and 0.722 on the RCT data set and GSSS data set, while YOLO returned an accuracy of 73.0% and 40.3% and IOU of 0.570 and 0.221, respectively (Table 1). Faster R-CNN returned an accuracy of 100% on 13 of the 18 species considered in the RCT data set, and 80% accuracy on 5 of the 11 species considering species with more than 100 images in the GSSS data set (Table 2 & 3). Figures 1-3 and 4-6 are examples of the Faster R-CNN performance for the RCT and GSSS data set respectively.

TABLE I
Comparison of Faster R-CNN and YOLO performance based on accuracy and IOU

Data Set	Model	Acc. (%)	IOU
RCT	Faster R-CNN	93.0 $\pm$ 3.20	0.80 $\pm$ 0.03
	YOLO	65.0 $\pm$ 12.1	0.57 $\pm$ 0.09
GSSS	Faster R-CNN	76.7 $\pm$ 8.31	0.72 $\pm$ 0.08
	YOLO	43.3 $\pm$ 14.5	0.22 $\pm$ 0.12

IV. Discussion

By utilizing modern approaches for object-detection, we demonstrate that researchers that require the analysis of

camera trap images can automate animal identification, quantification, and localization within images. Previous studies have demonstrated the quantification of animal individuals from camera trap data, but they suffer the limitation of returning a single classification per image, which is unrealistic for camera trap data. We demonstrate that Faster R-CNN is capable of accurately classifying more than one species per image given limited data when utilizing transfer learning.

Deep learning has demonstrated super-human performance on tasks with large amounts of data; however we test the reliability of deep learning methods on realistically sized ecological camera trap data sets. Without this distinction, deep learning approaches for autonomous camera trap data analysis would be limited to ecosystems with large numbers of labeled camera trap data, like Snapshot Serengeti, which required the effort of thousands of individuals to label. We demonstrate that if a research group performs a one-time labeling of less than 1,000 images, one can create a reliable model using Faster R-CNN. Our YOLO model performed poorly on both data sets, likely due to limited data.

While the GSSS data set contained approximately 4x the number of images, the trained model for the data set performed worse than the trained model for the RCT data set using Faster R-CNN. There are numerous explanations for this. First, the GSSS data set has extreme class imbalances, a well documented scenario where machine learning classifiers have had difficulty [32]. In addition, the GSSS data set is much 'messier' than the RCT data set, with the majority of images containing animals either extremely far away, cropped by the camera, obstructed behind another object/animal, and/or extremely close to the camera. While the RCT data set does contain some of these difficult scenarios, there are far fewer occurrences. When implementing models such as these, our results reiterate the importance of class balance. For real-life applications, if an animal of interest rarely appears in the camera trap data, we recommend finding and labeling additional images from outside sources to build a balanced data set, or exploring additional techniques for class imbalance.

Considering the success of the Faster R-CNN model, our method allows for future possibilities regarding detailed individual and behaviour analysis from camera trap images. Norouzzadeh et al. (2017) demonstrated this in its infancy by returning labeled classifications of young versus adult and male versus female classifications, and the specific behaviour found within the image [5]. This approach is not reliable, as if more then one species, age, sex, or behaviour are present, the classifier returns erroneous results. Object detection methods allow for the classifier to identify an age, sex, and behaviour of each individual within the image. Using this method of data collection, examples of autonomous ecological reports based on images with time-stamps include: comparing the movement patterns of genders within and across species, identifying seasonally when reproduction occurs by quantifying when infants are most active, and general comparisons of activity/behaviour across species, sex, and age.

While object detection provides promising steps forward, in order to reliably quantify population metrics, an automated system must be able to re-identify an individual it has previously seen. Camera trap re-identification methods suffer from an unavoidable bias when analyzed by a human and there is debate arguing against the reliability of humans when re-identifying animal individuals from camera trap data [33]. The development of a method for reliable animal re-identification would allow for autonomous population estimation of a given habitat using a formal mark and recapture model, such as Lincoln-Petersen [34]. Population estimates are reliant on accurate animal identification and if a deep learning system can demonstrate accurate animal re-identification, one could utilize these methodologies to create autonomous systems to extract a variety of ecological metrics, such as diversity, relative abundance distribution, and carrying capacity, contributing to larger overarching ecological interpretations of trophic interactions and population dynamics.

V. Conclusion

Recent advancements in the field of computer vision and deep learning have given rise to reliable methods of object detection. We demonstrated the successful training of an object detection classifier using the Faster R-CNN model considering limited ecological camera trap data. Utilizing object detection techniques, ecologists can now autonomously identify, quantify, and localize individual species within camera trap data without the previous limitation of returning only one species classification per image. Our findings show promising steps towards the automation of the labourious task of labeling camera trap images which can be used to improve our understanding of the population dynamics of ecosystems across the planet.

Acknowledgments

The authors would like to thank all the Snapshot Serengeti volunteers and the camera trap community at large for uploading their data for public access. We also thank the deep learning community at large for their continued pursuit of open sourcing new methodologies encouraging interdisciplinary works.

References

[1] L. W. Gysel and E. M. Davis, "A simple automatic photographic unit for wildlife research," *The Journal of Wildlife Management*, vol. 20, no. 4, pp. 451–453, 1956.

[2] K. U. Karanth, "Estimating tiger *Panthera tigris* populations from camera-trap data using capturerecapture models," *Biological conservation*, vol. 71, no. 3, pp. 333–338, 1995.

[3] A. C. Burton, E. Neilson, D. Moreira, A. Ladle, R. Steenweg, J. T. Fisher, E. Bayne, and S. Boutin, "Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes," *Journal of Applied Ecology*, vol. 52, no. 3, pp. 675–685, 2015.

[4] J. M. Rowcliffe and C. Carbone, "Surveys using camera traps: are we looking to a brighter future?" *Animal Conservation*, vol. 11, no. 3, pp. 185–186, 2008.

[5] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *ArXiv:1703.05830v5*, 2017.

[6] A. Chao, "Estimating population size for sparse data in capture-recapture experiments," *Biometrics*, pp. 427–438, 1989.

[7] X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, p. 52, 2013.

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

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position- neocognitron," *Electron. & Commun. Japan*, vol. 62, no. 10, pp. 11–18, 1979.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[17] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[18] H. Jeffries, M. Berman, A. Poularikas, C. Katsinis, I. Melas, K. Sherman, and L. Bivins, "Automated sizing, counting and identification of zooplankton by pattern recognition," *Marine biology*, vol. 78, no. 3, pp. 329–334, 1984.

[19] R. Simpson, P. Culverhouse, R. Ellis, and B. Williams, "Classification of euceratium gran. in neural networks," in *Neural Networks for Ocean Engineering, 1991., IEEE Conference on*. IEEE, 1991, pp. 223–229.

[20] H. Balfoort, J. Snoek, J. Smiths, L. Breedveld, J. Hofstraat, and J. Ringelberg, "Automatic identification of algae: neural network analysis of flow cytometric data," *Journal of Plankton Research*, vol. 14, no. 4, pp. 575–589, 1992.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[23] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 858–862.

[24] A. Gomez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: identification of animal species in camera-trap images using very deep convolutional neural networks," *arXiv preprint arXiv:1603.06169*, 2016.

[25] A. Gomez, G. Diez, A. Salazar, and A. Diaz, "Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds," in *International Symposium on Visual Computing*. Springer, 2016, pp. 747–756.

[26] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[27] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna," *Scientific data*, vol. 2, 2015.

[28] 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*. Springer, 2014, pp. 740–755.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] S. Nowozin, "Optimal decisions from probabilistic models: the intersection-over-union case," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 548–555.

[32] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[33] R. J. Foster and B. J. Harmsen, "A critique of density estimation from camera-trap data," *The Journal of Wildlife Management*, vol. 76, no. 2, pp. 224–236, 2012.

[34] D. Robson and H. Regier, "Sample size in Petersen mark–recapture experiments," *Transactions of the American Fisheries Society*, vol. 93, no. 3, pp. 215–226, 1964.

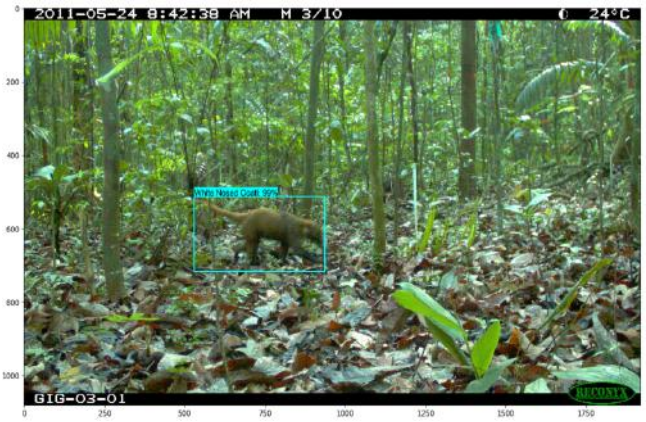

Fig. 1. Faster R-CNN output returning 1 White Nosed Agouti from the RCT data set in a highly camouflaged environment.

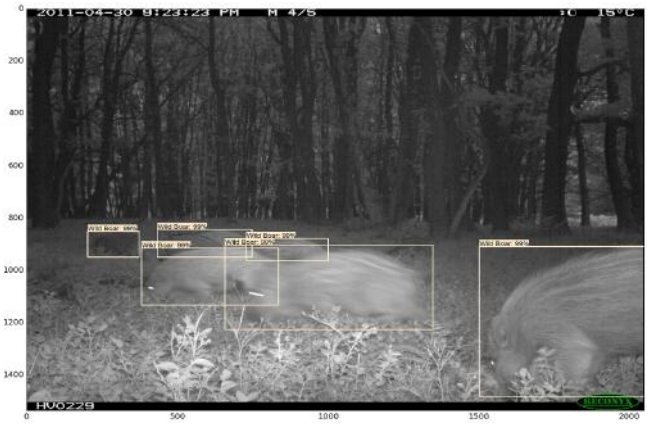

Fig. 2. Faster R-CNN output returning 6 wild boar classifications from the RCT data set in an image taken at night.

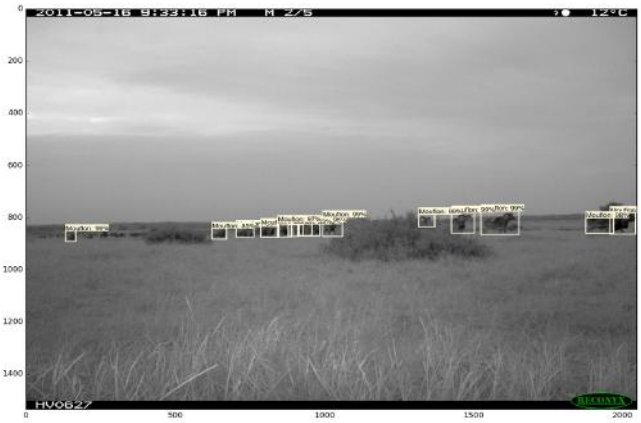

Fig. 3. Faster R-CNN output returning 15 Mouflon classifications from the RCT data set in an image taken at night.

Fig. 4. Faster R-CNN output returning 10 Wildebeest from the GSSS data set, demonstrating one example of the high levels of obstruction within the data set.

Fig. 5. Faster R-CNN output returning 4 Zebra and 4 Wildebeest from the GSSS data set, demonstrating two species within one image.

Fig. 6. Faster R-CNN output returning 10 Gazelle Thomsons from the GSSS data set, demonstrating the difficulties of distances.

TABLE II

RECONYX CAMERA TRAP (RCT) DATA SET DETAILED BREAK DOWN. FASTER R-CNN RETURNED AN AVERAGE ACCURACY OF 93.0% ACROSS ALL CLASSIFICATIONS. RCT CONTAINS A RELATIVELY EVEN CLASS DISTRIBUTION, LIKELY ATTRIBUTING TO FASTER R-CNN'S SUCCESS. THE STANDARD DEVIATIONS ARE QUITE HIGH DUE TO THE LIMITED NUMBER OF TESTING IMAGES WITHIN EACH CROSS-VALIDATION SET.

Reconyx Camera Trap					
Species	Scientific Name	Total Quantity	Total Images	Image Class Distribution (%)	Average Accuracy (%)
Mouflon	*Ovis orientalis orientalis*	126	45	4.8	100.0 ± 0.0
Collared Peccary	*Pecari tajacu*	96	82	8.7	71.4 ± 24.4
Agouti	*Dasyprocta*	87	87	9.2	91.7 ± 12.5
Wild Boar	*Sus scrofa*	81	56	5.9	100.0 ± 0.0
Red Deer	*Cervus elaphus*	68	68	7.2	100.0 ± 0.0
Red Brocket Deer	*Mazama americana*	63	63	6.7	100.0 ± 0.0
Ocelot	*Leopardus pardalis*	63	63	6.7	100.0 ± 0.0
White Nosed Couti	*Nasua narica*	60	38	4.0	100.0 ± 0.0
Paca	*Cuniculus*	57	57	6.0	100.0 ± 0.0
Great Tinamou	*Tinamus major*	52	44	4.6	50.0 ± 28.9
White Tailed Deer	*Odocoileus virginianus*	47	47	5.0	100.0 ± 0.0
Roe Deer	*Capreolus capreolus*	46	46	4.9	100.0 ± 0.0
Common Opossum	*Didelphis marsupialis*	44	44	4.6	100.0 ± 0.0
Red Squirrel	*Sciurus vulgaris*	39	39	4.1	66.7 ± 19.2
Bird Species	*Unlabeled*	38	29	3.1	100.0 ± 0.0
Spiny Rat	*Echimyidae*	34	34	3.6	88.9 ± 19.6
European Hare	*Lepus europaeus*	31	28	3.0	33.3 ± 38.6
Wood Mouse	*Apodemus sylvaticus*	29	29	3.1	100.0 ± 0.0
Red Fox	*Vulpes vulpes*	25	25	2.6	100.0 ± 0.0
Coiban Agouti	*Dasyprocta coibae*	23	23	2.4	50.0 ± 28.6

TABLE III

GOLD STANDARD SNAPSHOT SERENGETI (GSSS) DATA SET DETAILED BREAK DOWN. GSSS CONTAINS A HIGHLY IMBALANCED CLASS DISTRIBUTION LIKELY RELATED TO ITS POOR PERFORMANCE ACCURACY OUTSIDE OF A FEW MAIN CLASSIFICATIONS. FASTER R-CNN RETURNED AN AVERAGE ACCURACY OF 76.7% ACROSS ALL CLASSIFICATIONS.

Gold Standard Snapshot Serengeti

Species	Scientific Name	Total Quantity	Total Images	Image Class Distribution (%)	Accuracy (%)
Wildebeest	*Connochaetes*	11321	1610	40.0	89.1 ± 6.2
Zebra	*Equus quagga*	3677	767	18.9	61.7 ± 11.2
Buffalo	*Syncerus caffer*	987	227	6.00	37.0 ± 28.6
Gazelle Thomsons	*Eudorcas thomsonii*	938	198	4.88	92.0 ± 8.3
Impala	*Aepyceros melampus*	541	149	3.67	66.7 ± 19.2
Hartebeest	*Alcelaphus buselaphus*	351	242	5.96	80.0 ± 7.0
Guineafowl	*Numididae*	195	54	1.33	87.5 ± 8.6
Gazelle Grants	*Nanger granti*	176	61	1.50	12.0 ± 6.5
Warthog	*Phacochoerus africanus*	162	105	2.59	33.3 ± 14.6
Elephant	*Loxodonta*	125	85	2.10	50.0 ± 28.9
Giraffe	*Giraffa*	121	87	2.14	90.0 ± 12.7
Other Bird	*Unlabeled*	77	48	1.18	0.0 ± 0.0
Human	*Homo sapiens sapiens*	67	59	1.45	60.0 ± 14.6
Stork	*Ciconia ciconia*	63	12	0.296	50 ± 19.1
Spotted Hyena	*Crocuta crocuta*	62	54	1.33	50.0 ± 38.5
Eland	*Taurotragus oryx*	48	24	0.592	14.6 ± 19.2
Reedbuck	*Redunca*	44	29	0.715	66.7 ± 34.4
Oxpecker	*Buphagus*	43	14	0.345	0.0 ± 0.0
Baboon	*Papio*	35	22	0.542	14.3 ± 14.8
Lion	*Panthera leo*	34	17	0.419	8.4 ± 19.2
Hippopotamus	*Hippopotamus amphibius*	32	28	0.690	75.0 ± 14.0
Buff Crested Bustard	*Eupodotis gindiana*	27	15	0.370	0.0 ± 0.0
Topi	*Damaliscus korrigum*	24	16	0.394	0.0 ± 0.0
Cattle Egret	*Bubulcus ibis*	86	15	1.50	0.0 ± 0.0
Mongoose	*Herpestidae*	11	5	0.123	0.0 ± 0.0
Porcupine	*Hystrix africaeaustralis*	10	8	0.197	0.0 ± 0.0
Kori Bustard	*Ardeotis kori*	10	8	0.197	0.0 ± 0.0
Cheetah	*Acinonyx jubatus*	7	6	0.148	0.0 ± 0.0
Dik-dik	*Madoqua*	7	7	0.173	0.0 ± 0.0
Superb Starling	*Lamprotornis superbus*	6	3	0.0739	0.0 ± 0.0
Serval	*Leptailurus serval*	6	6	0.148	0.0 ± 0.0
Aardvark	*Orycteropus afer*	4	4	0.986	0.0 ± 0.0
Secretary Bird	*Sagittarius serpentarius*	4	4	0.0986	0.0 ± 0.0
Leopard	*Panthera pardus*	4	3	0.0739	0.0 ± 0.0
Buckbuck	*Tragelaphus sylvaticus*	4	4	0.0986	0.0 ± 0.0
Jackal	*Canis mesomelas*	3	3	0.0739	0.0 ± 0.0
Other Rodent	*Unlabeled*	3	1	0.0246	0.0 ± 0.0
Wattled Starling	*Creatophora cinerea*	3	1	0.0246	0.0 ± 0.0
Aardwolf	*Proteles cristata*	2	2	0.0493	0.0 ± 0.0
Ostrich	*Struthio camelus*	2	2	0.0493	0.0 ± 0.0
Hare	*Lepus microtis*	1	1	0.0246	0.0 ± 0.0
Grey Backed Fiscal	*Lanius excubitoroides*	1	1	0.0246	0.0 ± 0.0
Rhinoceros	*Rhinocerotidae*	1	1	0.0246	0.0 ± 0.0
Vervet Monkey	*Chlorocebus pygerythrus*	1	1	0.0246	0.0 ± 0.0
Waterbuck	*Kobus ellipsiprymnus*	1	1	0.0246	0.0 ± 0.0
White-Headed Buffalo Weaver	*Dinemellia dinemelli*	1	1	0.0246	0.0 ± 0.0