DOI: 10.21917/ijsc.2025.0516

A YOLOV4 METHOD FOR WILD ANIMAL COUNTING AND BEHAVIOUR DETECTION USING SMALL-SIZED CAMERA-TRAP IMAGE DATASET

Frank G. Kilima[1], Shubi Kaijage[2], Edith Luhanga[3] and Colin Torney[4]

[1,2,3]*School of Computational and Communications Sciences and Engineering, Nelson Mandela African Institution of Science and Technology, Tanzania*

[4]*School of Mathematics and Statistics, University of Glasgow, Scotland*

Abstract

Many deep learning-based solutions proposed to automate the analysis of camera-trap images for animal counting and behaviour detection deployed image classifiers that produce image-level labels, tackle animal counting as a classification task, and use images with one animal or attribute. They also used large image datasets which are costly, time-consuming and laborious to collect and annotate, not feasible for rare/elusive species and resource-constrained projects and did not explain the generalization of the models on images with untrained image backgrounds. This study developed animal counting and behaviour detection models for wild animals using You-Only-Look-Once (YOLO) and small-sized datasets with 20,110 camera-trap images. The study results show that using appropriate technologies including transfer learning, data augmentation and efficient data splitting methods, it is feasible to develop high-accurate and location-invariant object detection models using small-sized image datasets. Despite high performance, the animal counting model did not perform well on crowding/interacting animals including Guineafowl, Elephant, Lion, Zebra, Wildebeest, Baboon and Giraffe and it misclassified a significant number of Wildebeest as Buffalo and Zebra, but few Buffalo and Zebra were misclassified as Wildebeest. The behaviour detection model performed well on all behaviours except interacting. Model's poor performance on interacting is ascribed to its messy and small training set compared to other behaviour classes. The selection of an optimal confidence threshold appropriate for a particular dataset and increasing data diversity of the training set can significantly improve recall while reducing false positives.

Keywords:

Artificial neural network (ANN), Convolutional neural network (CNN), Deep learning, Generalization, Object detection, You-Only-Look-Once (YOLO)

1. INTRODUCTION

Increased activities and phenomena including environmental degradation, decline of biodiversity, poaching, over-harvesting, wildlife diseases, rapid loss of habitat, changing patterns of land use, and climate change have led to various adverse consequences on wildlife. These include the decline of wildlife populations, increase of human-wildlife conflicts and even extinction of some wildlife animal species [1]–[4]. These consequences have necessitated frequent studies on wildlife populations, abundances, density, habitat use, intra-community interactions and structure, animal distribution, ecosystems, population dynamics, monitoring, and behaviours [5]–[7] using various wildlife data collected via different methods. Camera-trapping is a commonly used method for collecting imagery data for many such studies owing to its benefits which include inexpensiveness, ease of deployment and maintenance, non-obtrusiveness to animals and safety to both animals and project staff. Its other benefits include minimal human involvement, producing high-quality and permanent records, operating well in hard-to-access areas, dangerous areas, day and night, and for long durations as battery and memory can permit [1], [7]–[12].

Because of its massive use, camera-trapping generates a copious number of camera-trap images. For instance, operating in Serengeti National Park (SNP) in Tanzania, the Snapshot Serengeti (SS) Project and Serengeti Biodiversity Program (SBP) generated about 7.1 million and 1 million camera-trap images, respectively [13], [14]. To reduce time, labour, cost, and reliance on domain expertise inherent in the manual analysis of collected camera-trap images, many wildlife studies have automated the analysis by deploying machine learning/deep learning (ML/DL) methods. However, despite this great success, the use of image classifiers such as ResNet, VGG etc. by many proposed automated solutions including Yousif et al., (2019), Chen et al., (2015), Norouzzadeh et al., (2018), Tabak et al., (2019) led to several limitations. First, image classifiers produce image-level labels regardless of the number of animals or attributes the image contains. Second, image classifiers tackle animal counting as a classification task [7], and many solutions use animal images with one animal or attribute (behaviour) [1], [7], [10]. This leads to inaccurate results specifically when analysing images with multiple animals (attributes). Third, many solutions such solutions use large datasets with hundreds of thousands or millions of camera-trap images [7], [10], [12], [15]. Such large datasets are costly, laborious, time-consuming and computationally expensive to collect, annotate and develop DL models. It is also challenging to collect large datasets for rare and elusive animal species or resource-constrained projects. Fourth, many such studies estimate the performance of the developed models on trained image backgrounds [1], [7] i.e., did not estimate models' performance on images with untrained backgrounds. Last, owing to insufficient literature, further research is to be done provide insight into the performance of object detection models on behaviour detection in camera-trap images.

This study used a YOLOv4 object detector and 20,110 camera-trap images to develop an animal counting model for 11 wild animal species (Table 1), and a behaviour detection model for five animal behaviours (Table 2). This study offers several contributions including the use of small-sized datasets (which reduce cost, time, labour and computational power required to collect, annotate and develop models). The use of small-sized datasets promotes the development of effective DL solutions for resource (images) constrained wildlife studies. The validation of trained models on images with untrained image backgrounds provides insight into the generalization of object detection models on untrained image backgrounds in comparison with their generalization on trained image backgrounds. To further describe and understand the model's performance on individual and overall classes, this study deployed several performance metrics.

Lastly, since collecting and annotating images for object detection tasks is costly, laborious and time-consuming, the study contributes two labelled datasets for animal detection and behaviour detection which can be used by other researchers for similar studies.

2. OBJECT DETECTION AND YOLOV4

Object detection is the task that involves locating and classifying objects of interest within an image or video [16], [17]. According to Gündüz & Işık, (2023), object detection comprises object localization; a process of predicting the location of an object in an image and drawing a bounding box around it; and object classification, which is the process of predicting the class to which the predicted object belongs. Two major categories of convolutional neural network-based object detection algorithms are one-stage object detection algorithms and two-stage-object detection algorithms [17], [18]. Three sequential operations (steps) of two-stage object detection algorithms involve generating bounding box candidates, object localization and object classification. On another hand, one-stage object detection algorithms perform object localization and object classification for the complete image simultaneously, making them generally faster, but less accurate than two-stage object detection algorithms [3], [16], [17]. Examples of one-stage object detection algorithms are Single Shot MultiBox Detector (SSD), RetinaNet, Fully Convolutional One-Stage (FCOS), and You-Only-Look-Once (YOLO) series algorithms (including YOLOv4). Popular examples of two-stage object detection algorithms are Region with Convolutional Neural Network (R-CNN), Fast R-CNN, Faster R-CNN, Region-based Fully Convolutional Network (R-FCN) and Libra R-CNN object detectors [3], [16].

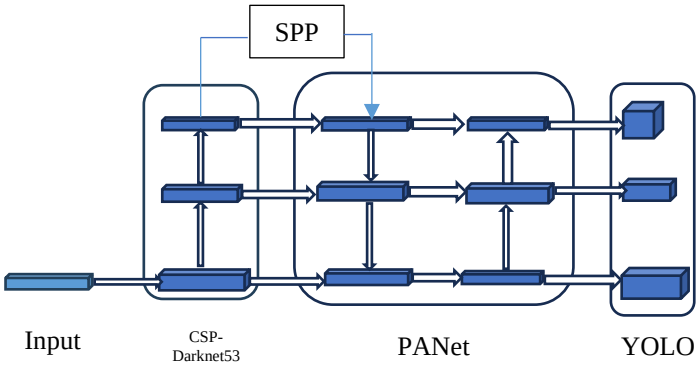

Fig.1. The architecture of YOLOv4 (Source: [20])

YOLOv4 is the fourth version in the family of YOLO object detection algorithms which was first proposed by Alexey Bochkovskiy and published in April 2020 [16], [19] after being tested extensively on MS COCO dataset. YOLOv4 offers significant improvement in inference time (speed) over YOLOv3. It is also implemented on an open-source, flexible and high-performance framework called Darknet, capable of being easily implemented on both a central processing unit (CPU) and a graphics processing unit (GPU). Because of its high processing speed and comparatively good detection performance, YOLO is a preferred choice among a stack of deep learning object detection algorithms for real-time object detection tasks including self-driving cars, traffic surveillance systems, and smart self-governance [19]–[21]. To localize and classify objects in an image, YOLOv4 divides an image into several grids, with the bounding box and confidence score for each grid being predicted. Finally, the class for each detected object is predicted based on the highest confidence score [16]. YOLOv4 architecture consists of three major components namely backbone, detection neck, and detection head [16], [17]. It also uses Cross Stage Partial Darknet53 (CSPDarknet53) which is a convolutional neural network (CNN) for the backbone, Spatial Pyramid Pooling (SPP) Block and Path Aggregation Network (PANet) for the detection neck and three YOLOv3 heads with sizes [17] for detection head [16], [17], [20].

3. MATERIALS AND METHODS

3.1 DATA ACQUISITION AND SPLITTING

This study used 20,110 camera-trap images from the SS project and SBP. 13,693 images were used for animal counting task, while 6417 images for behaviour detection task. 13,170 images of the animal counting task were from the SS dataset, while 523 images were from SBP. This dataset was split under a single holdout cross-validation method with stratified random sampling technique into 8815 images (64%), 2204 images (16%) and 2674 images (20%) for training set, validation set and test set respectively. The test set was further split into two distinct subsets which are the trained test (1338 images), and the untrained test set (1336 images). Training set shared image background domain with the trained test set. However, this background domain was different from that of untrained test set images.

Table.1. Dataset split for animal counting task

Species	Training set	Validation set	Trained set	Untrained set	Total
Buffalo	642	160	101	101	1004
Elephant	744	186	110	111	1151
Giraffe	920	230	146	144	1440
Guineafowl	931	233	140	139	1443
Hyena	1081	270	155	158	1664
Lion	862	215	125	125	1327
Hartebeest	479	120	75	74	748
Warthog	979	246	146	144	1515
Wildebeest	575	144	95	94	908
Zebra	1351	338	200	200	2089
Baboon	251	62	45	46	404

The development of the behaviour detection model utilized a dataset comprising 6417 images representing five animal behaviours: feeding, moving, resting, standing and interacting. The dataset was split using a single holdout cross-validation method with stratified random sampling technique into 4604 images (72%), 1151 images (18%) and 662 images (10%) for training set, validation set and test set, respectively. The breakdown of this dataset is shown in Table 2. All images for animal counting and behaviour detection tasks were hand-labelled (annotated) into YOLO format (.txt) using a labelling software called labeling.

Table.2. Single holdout cross validation scheme for behaviour detection

Behaviour class	Training set	Validation Set	Test set	Total
FEEDING	920	230	144	1294
MOVING	1312	328	182	1822
RESTING	872	218	122	1212
STANDING	1104	276	157	1537
interacting	396	99	57	552
TOTAL	4604	1151	662	6417

3.2 ENVIRONMENT SETUP

All animal counting and behaviour detection models were developed on Google Colab Pro Tesla T4 GPU with 15 GB of RAM in the darknet framework. All training and validation images were automatically resized into 416x416 pixels (width x height). The study deployed transfer learning by using models pre-trained on the Microsoft Common Object in Context (MS COCO) dataset and data augmentation techniques. The two techniques are commonly deployed in developing ML/DL models in situations with less training data and imbalanced classes to enhance fast convergence, improve models' performance and generalization and reduce overfitting, training time and computational power [9], [22]–[24]. Although MS COCO has fewer images and object categories than the ImageNet dataset, it is more appropriate for transfer learning for this study than ImageNet because it contains fewer iconic objects, more object categories and object instances per image, fewer images with one object category and several classes [25]–[27].

3.3 EVALUATION METRICS

To assess the performance of the developed models, we used eight different evaluation metrics namely true positive (TP), false positive (FP), false negative (FN), recall, precision, F1-score, average precision (AP), and mean average precision (mAP). A true positive refers to the detection of a positive ground-truth bounding (box) sample [28], [29]. A false positive is defined as an incorrect detection of a non-existent object or misclassification of an existing object in an image [28], [29]. A false negative is defined as an undetected ground-truth bounding box (sample) [28], [29], or a detected but misclassified ground-truth sample.

Recall is the ability of the object detector to identify all relevant samples of the ground truth [28], [29]. It is expressed as a proportion of true positives over all ground-truth samples, i.e., $\frac{TPs}{TPs + FNs}$. Precision is defined as the ability of an object detector to detect only relevant samples among positive detections [28], [29]. It is expressed as the proportion of true positive overall positive detections made by an object detector, i.e., $\frac{TPs}{TPs + FPs}$. F1-score is defined as a harmonic mean of precision and recall [22], [29] expressed as $2 \text{ x} \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}}$.

According to Schneider et al., (2020), an F1-score of at least 0.7 indicates that the model attained better recall and precision values. Average precision is an evaluation metric which expresses the percentage of correct predictions of a particular class in the dataset obtained by computing an area under curve (AUC) of precision x recall curve and averaging precision across all recall values between 0 and 1 at one or various intersection over union (IoU) values [28]. The mAP is an evaluation metric which expresses the model's accuracy over all classes in the dataset, computed by averaging average precisions of all classes in a dataset into a single numerical performance [28], [29].

4. RESULTS

4.1 ANIMAL COUNTING

Before estimating the model's accuracy on animal counting, it is important to estimate its accuracy in detecting and classifying animals in the given test set. The Table.3 presents the model's accuracy in AP and mAP on the detection and classification of animals on trained and untrained test sets. The accuracy results show a slight difference of 2.18% in mAP of the model's accuracy between trained and untrained test sets. Results further show that the model performed well in all animal classes on both test sets, except on Baboon, Wildebeest and Buffalo. They also show that the animal counting model performed better on the trained test set than on the untrained test set on all animal classes except on Buffalo, Guineafowl, Hartebeest and Baboon and that there exists a small performance difference in AP between the two test sets on all classes except Giraffe (7.59%) and Lion (12.48%). To further describe its performance, the model's accuracy in counting detected animals in terms of true positives (TPs), false positives (FPs), recall, precision, F1-score was estimated using an untrained test set with 3718 ground-truth (GT) samples. The results from this estimation are presented in Table 4 and Table 5.

Table.3. Model's detection and classification accuracy on trained and untrained test sets

Animal Class	AP on trained Test set (%)	AP on untrained Test set (%)
Buffalo	69.81	70.10
Elephant	82.59	81.87
Giraffe	96.13	88.54
Guineafowl	90.52	91.40
Hyena	98.17	95.71
Lion	92.86	80.38
Hartebeest	82.90	86.22
Warthog	95.82	91.93
Wildebeest	71.97	71.85
Zebra	79.12	71.01
Baboon	71.63	72.36
mAP (%)	**84.68**	**82.50**

Table.4. Model's accuracy on counting detected animals on untrained test set

Animal class	GT	TPs	FPs	FNs	Recall (%)	Precision (%)	Total Counts
Buffalo	449	315	92	134	70.16	77.40	407
Elephant	300	242	11	58	80.67	95.65	253
Giraffe	205	175	1	30	85.37	99.43	176
Guineafowl	510	427	3	83	83.73	99.30	430
Hyena	164	150	1	14	91.46	99.34	151
Lion	205	157	9	48	76.59	94.58	166
Hartebeest	126	110	9	16	87.30	92.44	119
Warthog	193	176	6	17	91.19	96.70	182
Wildebeest	655	407	7	248	62.14	98.31	414
Zebra	829	620	25	209	74.79	96.12	645
Baboon	82	56	3	26	68.29	94.92	59
Total/average	**3718**	**2835**	**167**	**883**	**79.24**	**94.93**	**3002**

The model attained the mean (average) F1-score of 86.10%. Table 4 and Table 5 show that the animal counting model made a total of 3002 animal counts, of which 2835 were TP counts and 167 FP counts, therefore attaining an average recall of 79.24%, average precision of 94.93% and average F1-score of 86.10%. Table 5 and S1 (Supplemental materials) show that of the 167 FP counts, 111 were trained animals, 12 were untrained objects (rock, tree, wall, untrained animals and log) and 44 (of which 36 were false positive Buffalo) were a result of double detections. Double detections involve multiple detections of the same object in an image. When this occurs, the detection with the highest confidence score is compared against an object's ground-truth bounding box to determine whether it is a TP or not, while the other detections with lower confidence scores are considered FP [30].

Table.5. Breakdown of false positive and false negative detections

Animal class	False Positive Detections				False Negative Detections		
	Trained animals	Untrained Animals	Double Detections	Total FP	Misclassified Detections	Undetected Detections	Total FNs
Buffalo	55	1	36	92	15	119	134
Elephant	10	0	1	11	2	56	58
Giraffe	0	1	0	1	0	30	30
Guineafowl	0	3	0	3	0	83	83
Hyena	1	0	0	1	8	6	14
Lion	8	1	0	9	8	40	48
Hartebeest	8	1	0	9	3	13	16
Warthog	4	0	2	6	7	10	17
Wildebeest	4	2	1	7	64	184	248
Zebra	19	3	3	25	3	206	209
Baboon	2	0	1	3	1	25	26
Total	111	12	44	167	111	772	883

The Table.4 further shows that there are 883 false negatives (23.75% of the GT samples), of which 772 were unidentified GT samples while 111 were identified but misclassified GT samples. Results further show that Buffalo (92 FPs), Zebra (25 FPs) and Elephant (11 FPs) accounted for 76.65%) of all 167 FP detections, while Wildebeest (184), Zebra (206), Buffalo (119), Guineafowl (83), Elephant (56), and Lion (40) accounted for about 89.12% of all 772 unidentified GT samples. Table 5 shows that Wildebeest accounted for 64 detections (57.66%) of all 111 identified but misclassified GT samples.

4.2 BEHAVIOUR DETECTION

The Table.6 presents the accuracy (in APs and mAPs) of the best-performing behaviour detection model on the test set. The Table shows that the model attained mAP of 69.55% and that it performed well on feeding, moving and standing, fairly on standing, and poorly on interacting.

Table.6. Accuracy

	Behaviour class					mAP (%)
	A	B	C	D	E	
AP (%)	72.51	77.83	76.22	65.37	55.84	69.55

Key: A = feeding, B = moving, C = resting, D = standing, E= interacting

5. DISCUSSION

Owing to the high accuracy attained, this study demonstrates the feasibility of developing accurate object detection solutions for wild animal counting and behaviour detection using small-sized datasets in combination with techniques including data augmentation, transfer learning, efficient data splitting, hyperparameter tuning etc. This advantage renders an opportunity for resource-constrained or rare animal projects which cannot collect large camera-trap image datasets to also significantly benefit from deep learning methods. The use of smaller datasets reduces the amount of time, money, labour and computational power required to collect and annotate large datasets and develop object detection models. Results also show that object detection solutions can generalize well to new images with untrained backgrounds and are therefore appropriate for the development of location-invariant detection models for animal counting and behaviour detection. Like the use of small-sized datasets, location-invariant object detection models will render the re-use (redeployment) of existing object detection solutions and therefore save money, time, labour and computational power spent on developing camera-trap location variant (specific) solutions. The model's performance results show that animal counting accuracy is inextricably tied to its ability to identify relevant animals from the ground-truth samples (recall) and classify positive detections as true positives (precision). As indicated in Table 5, although the animal counting model attained high recall (79.24%), still it did not identify 772 samples (20.76%) of ground-truth samples, which is numerically large compared to

111 ground-truth samples which were identified but misclassified. Results from Table 5 and further observation of the detection outputs show that a large number of unidentified ground-truth samples (by the proportion of unidentified samples to ground-truth samples for each animal class) were from Giraffe (14.6%), Guineafowl (16.3%), Elephant (18.7%), Lion (19.5%), Zebra (24.8%), Buffalo (26.5%), Wildebeest (28.1%) and Baboon (30.5%). Most of these species have the tendency to live in crowds (concentrated groups) or interact physically, making it harder for the model to easily detect them individually. In addition, the animal counting model identified a significant number of Baboons carrying young Baboons or two Giraffes moving or standing close to each other (playing or fighting) as one animal. Similarly, observations on unidentified samples revealed that the animal counting model did not identify flying or wing-stretching Guineafowl samples as well as Baboons sitting on trees, which may be ascribed to a lack of images with such features in the training set. These observations suggest that adding images with such features in the training set can significantly improve the model's recall. From literature, much of the emphasis is put on the use of large training sets rather than wider feature diversity. However, it is possible to use large training sets with less (similar) feature diversity and therefore attain low recall. This may occur when a resource-constrained project, owing to the high cost associated with the labelling of images and developing the model, decides to use few of the available images.

Further, results (supplemental material S1) have shown that the animal counting model misclassified Wildebeest as Buffalo and Zebra more than Buffalo and Zebra misclassified as Wildebeest. While resemblance (skin colour, shape, horns etc.) between Wildebeest and Buffalo may be ascribed to these misclassifications, it does not explain why there are few misclassifications of Buffalo as Wildebeests. Similarly, less resemblance between Wildebeest and Zebra creates questions for factors leading to many misclassifications of Wildebeests as Zebra but zero misclassified Zebra as Wildebeest (supplemental material S1). Poor performance of the behaviour detection model on interacting behaviour may be ascribed to messy and small-size training data of the interacting behaviour class. The class contained the smallest proportion (8.6%) of the behaviour detection dataset, which is 2.2 and 3.3 times smaller than RESTING (second smallest behaviour class) and MOVING (largest behaviour class). It also contained messy data, largely composed of crowded animals including Lions, Zebra, Elephants, Wildebeest and Baboons. Similarly, the animal counting model produced a significant number of unidentified ground-truth samples (false negatives) from these species owing to their crowdy nature in many images.

Additionally, it was observed that a significant number of instances of interacting images were not as conspicuous as of other behaviours including resting, standing, feeding and moving. A combination of these factors made it more challenging for the behaviour detection model to identify instances of interacting behaviour than it was with other behaviours. We also observed that the selection of an optimal confidence threshold may significantly improve the model's recall by reducing the number of unidentified ground-truth samples (false negatives) from images.

Our study demonstrates that a confidence threshold of 0.5 (or above) is too restrictive and produces many unidentified false negatives. It also demonstrates that a confidence threshold value below 0.4 though significantly reduces unidentified false negative samples, also significantly increases the number of positively detected but misclassified samples and double detections. We established that an optimal confidence threshold lies between 0.4 and 0.5.

6. CONCLUSION AND AREAS FOR FURTHER STUDIES

This study used a YOLOv4 object detector and small-sized camera-trap image datasets to develop accurate detection models for wild animal counting and behaviour detection in combination with techniques including transfer learning, data augmentation and efficient data splitting. Study results further demonstrate that object detection models can generalize well on untrained image backgrounds and therefore used to develop location-invariant object detection models. This advantage is useful for resource-constrained wildlife projects which cannot collect large camera-trap datasets or studies for rare or elusive species which commonly have few camera-trap images. Development of location-invariant object detection models using small-sized image datasets saves time, money, effort and computational power commonly spent on developing location-variant object detection solutions using large datasets.

Our study observed that the animal counting model misclassified more Wildebeests as Buffalo and Zebra than misclassified Buffalo and Zebra as Wildebeests, combined and that the model did not perform well on crowded animal species including Zebra, Wildebeest, Guineafowl, Elephant and Buffalo. We also found that double detections and false positives from untrained animals have insignificant effects on animal detection and counting. However, the study results postulate that a small training set and messy data are ascribed to poor performance of behaviour detection model for interacting behaviour. Given more images and of high quality, the model would perform better.

Our study results show that improving feature diversity in the training set may significantly improve the recall of the detection models. We recommend further research on the factors influencing the misclassification of more wildebeest as buffalo and zebra, compared to buffalo and zebra being misclassified as wildebeest, methods to improve detection of crowded (concentrated) animals including Lions, Zebra, Elephants, Wildebeest and Baboons. Crowded species have the largest proportion of unidentified GT samples, and therefore high FNs. Further model training on dataset with increased size and quality of interacting images, and with flying birds may significantly improve models' recall. We also recommend the use of untrained images (images with new backgrounds) from geographically different camera-trap locations as test set to further study the generalization of the object detection models on untrained image backgrounds.

SUPPLEMENTAL MATERIAL

Table.7. Confusion matrix of the test set

		Buffalo	Elephant	Giraffe	Guineafowl	Hyena	Lion	Hartebeest	Warthog	Wildebeest	Zebra	Baboon	Detected FPs
Animal class		**FPs from trained animals**											
Ground Truth	Buffalo		8					1	1	2	3		15
	Elephant	2											2
	Giraffe												
	Guineafowl												
	Hyena	2					4		1			1	8
	Lion		2			1		3	1	1			8
	Hartebeest						1			1	1		3
	Warthog	3					3					1	7
	Wildebeest	45						4			15		64
	Zebra	3											3
	Baboon								1				1
Subtotal from GT		**55**	**10**	**0**	**0**	**1**	**8**	**8**	**4**	**4**	**19**	**2**	**111**
Animal Class		**FPs from double detections**											
Ground Truths	Buffalo									1			1
	Elephant	2											2
	Giraffe												
	Guineafowl												
	Hyena								2			1	3
	Lion		1										1
	Hartebeest												
	Warthog	10											10
	Wildebeest	24									3		27
	Zebra												
	Baboon												
Subtotal FPs from double detections		**36**	**1**	**0**	**0**	**0**	**0**	**0**	**2**	**1**	**3**	**1**	**44**
Untrained objects		**FPs from untrained objects**											
Untrained	Rock				3					2			5
	Tree	1		1							2		4
	Wall							1					1
	Untrained animals										1		1
	Log						1						1
Subtotal FPs from untrained		**1**	**0**	**1**	**3**	**0**	**1**	**1**	**0**	**2**	**3**	**0**	**12**
Total FPs (FPs from GT + FPs from double detections + FPs from untrained objects)													
Total detections		**92**	**11**	**1**	**3**	**1**	**9**	**9**	**6**	**7**	**25**	**3**	**167**

REFERENCES

[1] G. Chen, T.X. Han, Z. He, R. Kays and T. Forrester, "Deep Convolutional Neural Network based Species Recognition for Wild Animal Monitoring", *Proceedings of International Conference on Image Processing*, pp. 1-6, 2015.

[2] C.J. Torney, "A Comparison of Deep Learning and Citizen Science Techniques for Counting Wildlife in Aerial Survey Images", *Methods in Ecology and Evolution*, Vol. 10, No. 6, pp. 779-787, 2019.

[3] Y. Li, "A Deep Learning-based Hybrid Framework for Object Detection and Recognition in Autonomous Driving", *IEEE Access*, Vol. 8, pp. 194228-194239, 2020, 2020.

[4] D. Wang, Q. Shao and H. Yue, "Surveying Wild Animals from Satellites, Manned Aircraft and Unmanned Aerial Systems: A Review", *Remote Sensing*, Vol. 11, No. 11, pp. 1-6, 2019.

[5] A.C. Burton, "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.

[6] R. Amin, A.E. Bowkett and T. Wacher, "The Use of Camera-Traps to Monitor Forest Antelope Species", *Antelope Conservation from Diagnosis to Action*, pp. 190-216, 2016.

[7] M. S. Norouzzadeh, "Automatically Identifying, Counting and Describing Wild Animals in Camera-Trap Images with Deep Learning", *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 115, No. 25, pp. 5716-5725, 2018.

[8] 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*, Vol. 12, No. 1, pp. 150-161, 2021.

[9] S. Schneider, G.W. Taylor and S. Kremer, "Deep Learning Object Detection Methods for Ecological Camera Trap Data", *Proceedings of International Conference on Computer and Robot Vision*, pp. 321-328, 2018.

[10] M.A. Tabak, "Machine Learning to Classify Animal Species in Camera Trap Images: Applications in Ecology", *Methods in Ecology and Evolution*, Vol. 10, No. 4, pp. 585-590, 2019.

[11] B.G. Weinstein, "A Computer Vision for Animal Ecology", *Journal of Animal Ecology*, pp. 533-545, 2017.

[12] 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*, Vol. 9, No. 4, pp. 1578-1589, 2019.

[13] 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*, pp. 1-14, 2015.

[14] A.S. Mohammed and V. Mallikarjunaradhya, "Optimizing Real-time Task Scheduling in Cloud-based AI Systems using Genetic Algorithms", *Proceedings of International Conference on Contemporary Computing and Informatics*, Vol. 7, pp. 1649-1653, 2024.

[15] A.G. Villa, A. Salazar and F. Vargas, "Ecological Informatics Towards automatic Wild Animal Monitoring : Identification of Animal Species in Camera-Trap Images using Very Deep Convolutional Neural Networks", *Ecology Informatics*, Vol. 41, pp. 24-32, 2017.

[16] M.Ş. Gündüz and G. Işık, "A New YOLO-based Method for Social Distancing from Real-Time Videos", *Neural Computing and Applications*, Vol. 3, pp. 1-6, 2023.

[17] X. Zhang, K. Eltouny, X. Liang and S. Behdad, "Automatic Screw Detection and Tool Recommendation System for Robotic Disassembly", *Journal of Manufacturing Science and Engineering*, Vol. 145, No. 3, 2023.

[18] J. Jia, M. Fu, X. Liu and B. Zheng, "Underwater Object Detection based on Improved EfficientDet", *Remote Sensing*, Vol. 14, No. 18, pp. 1-6, 2022.

[19] R.H. Chang, Y.T. Peng, S. Choi and C. Cai, "Applying Artificial Intelligence (AI) to Improve Fire Response Activities", *Emergency Management Science and Technology*, Vol. 2, No. 1, pp. 1-6, 2022.

[20] J. Woo, J.H. Baek, S.H. Jo, S.Y. Kim and J.H. Jeong, "A Study on Object Detection Performance of YOLOv4 for Autonomous Driving of Tram", *Sensors*, Vol. 22, No. 22, pp. 1-7, 2022.

[21] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan and Z. Mahmood, "A Fast and Accurate Real-Time Vehicle Detection Method using Deep Learning for Unconstrained Environments", *Applied Sciences*, Vol. 13, No. 5, pp. 1-6, 2023.

[22] S. Schneider, S. Greenberg, G.W. Taylor and S.C. Kremer, "Three Critical Factors Affecting Automated Image Species Recognition Performance for Camera Traps", *Ecology and Evolution*, Vol. 10, No. 7, pp. 3503-3517, 2020.

[23] J. LeBien, "A Pipeline for Identification of Bird and Frog Species in Tropical Soundscape Recordings using a Convolutional Neural Network", *Ecological Informatics*, Vol. 59, pp. 1-6, 2020.

[24] V. Maeda-Gutiérrez, "Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases", *Applied Sciences*, Vol. 10, No. 4, pp. 1-6, 2020.

[25] L. Jiao, "A Survey of Deep Learning-based Object Detection", *IEEE Access*, Vol. 7, No. 3, pp. 128837-128868, 2019.

[26] T. Y. Lin, "Microsoft COCO: Common Objects in Context", *Lecture Notes in Computer Science*, Vol. 8693, pp. 740-755, 2014.

[27] Z.Q. Zhao, P. Zheng, S.T. Xu and X. Wu, "Object Detection with Deep Learning: A Review", *IEEE Transactions on Neural Networks Learning Systems*, Vol. 30, No. 11, pp. 3212-3232, 2019.

[28] R. Padilla, S.L. Netto and E.A.B. Da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms", *Proceedings of International Conference on Systems, Signals and Image Processing*, pp. 237-242, 2020.

[29] R. Padilla, W.L. Passos, T.L.B. Dias, S.L. Netto and E.A.B. Da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit", *Electronics*, Vol. 10, No. 3, pp. 1-28, 2021.

[30] S. Agarwal, J.O. Du Terrail and F. Jurie, "Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks", *Computer Science*, pp. 1-6, 2019.