IAES International Journal of Robotics and Automation (IJRA)
Vol. 10, No. 2, June 2021, pp. 133~143
ISSN: 2722-2586, DOI: 10.11591/ijra.v10i2.pp133-143

Detecting African hoofed animals in aerial imagery using convolutional neural network

Yunfei Fang[1], Shengzhi Du[2], Larbi Boubchir[3], Karim Djouani[4]
[1,2,4]Department of Electrical Engineering, Tshwane University of Technology, South Africa
[3]LIASD, University of Paris 8, France

Article Info

Article history:

Received Aug 31, 2020
Revised Dec 1, 2020
Accepted Feb 12, 2021

Keywords:

Anchor design
Animal detection
Atrous convolution
Faster R-CNN
Small object detection

ABSTRACT

Small unmanned aerial vehicles applications had erupted in many fields including conservation management. Automatic object detection methods for such aerial imagery were in high demand to facilitate more efficient and economical wildlife management and research. This paper aimed to detect hoofed animals in aerial images taken from a quad-rotor in Southern Africa. Objects captured in this way were small both in absolute pixels and from an object-to-image ratio point of view, which were not perfectly suit for general purposed object detectors. We proposed a method based on the iconic Faster region-based convolutional neural networks (R-CNN) framework with atrous convolution layers in order to retain the spatial resolution of the feature map to detect small objects. A good choice of anchors was of prime importance in detecting small objects. The performance of the proposed Faster R-CNN with atrous convolutional filters in the backbone network was proven to be outstanding in our scenario by comparing to other object detection architectures.

Corresponding Author:

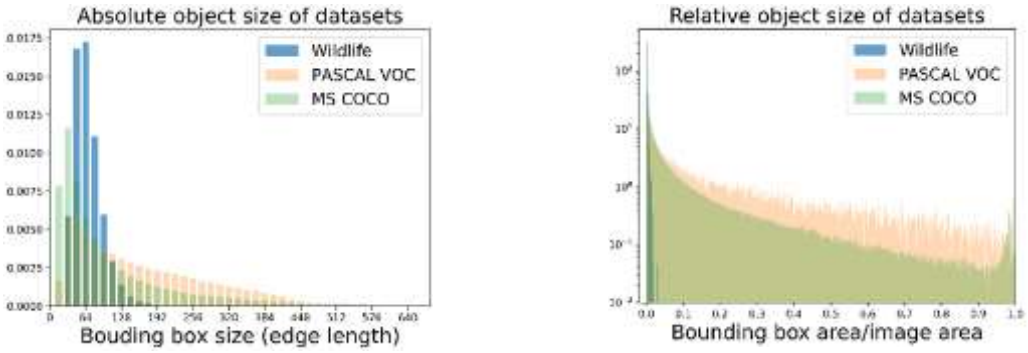
Yunfei Fang
Department of Electrical Engineering
Tshwane University of Technology
Staatsartillerie Road, Pretoria west, Pretoria, 0001, South Africa
Email: fangyunfei08@163.com

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) as a convenient and easy-to-get data acquisition tool has been applied in many wildlife conservation and research tasks. Automatic analysis of such aerial imagery is of significant importance as such data amounts dramatically. Object detection forms the basis of many computer vision applications. Work has been done in detecting terrestrial [1], marine [2], and celestial [3, 4] species from aerial imagery in different environments. Africa holds a variety of unique hoofed wildlife species, and a number of them are under threat, some at critical risk of extinction. UAVs combined with computer vision techniques can assist conservation workers and researchers to a great extent.

Computer vision applications differ from case to case depending on the scenario and unique characteristics of the provided dataset. Challenges of object detection in aerial imagery were summarised in [5], such as small object size, large scale variations, crowded instances and various orientations. In among which the most challenging is the size and scale problem. The UAV has to operate at a certain altitude in order to provide a big field of view, avoid disturbing the targeted and other local species. The distance makes the animals captured in small scale. The full image resolution should be big enough to retain the absolute object size in pixels so that it carries sufficient information. This makes the ratio of the object size to the full image size quite small compared to objects in ground-perceived imagery.

To illustrate this, summaries of the object size measured by the bounding-box areas are given in Figure 1. Our dataset was compared to the popular generic object detection datasets PASCAL-VOC [6] and MS-COCO [7], which contains everyday objects such as cars, pedestrians and pets. Figure 1 (a) shows the histogram of the absolute object size in pixels. Object size was represented by the square root of the bounding-box area. The bins are normalized to compare the datasets on the same scale. Majority of the objects in our wildlife dataset lies in edge length no bigger than 128 pixels. While the object size of our dataset is smaller than the other datasets, the full image size is much bigger. This makes the ratio of the object size against the full image size even smaller, as shown on Figure 1 (b). Figure 2 shows some example zebras cropped from the aerial imagery we captured from the wild nature. Figure 2 (a) shows the various orientations of the zebras from the bird's eye view. In Figure 2 (b), the zebras were found difficult to be distinguished from other animals due to the small scale and illumination conditions.



(a) (b)

Figure 1. Characteristics of object size in our dataset, PASCAL−VOC, and MS−COCO: histogram of (a) absolute object size in edge length and (b) relative object size (object size/image size), showing objects in our dataset occupies only little portion of the full image

(a)

(b)

Figure 2. Example of zebras cropped from our dataset: (a) different orientations of the zebras from aerial perspective and (b) zebras in distance under various illumination

Deep learning frameworks such as recurrent neural network (RNN) [8] and convolutional neural network (CNN) [9] have boosted machine learning applications to another level in recent years. Research in fields like nature language processing (NLP) [10], machine translation [11], and computer vision [12, 13] have been dominated by deep learning. The word "deep" refers to the depth of information that the neural network extracts from the raw data. Modern CNN facilitated object detectors derive the feature representations by stacking a sequence of convolutional layers over the input image. During the process, the feature map size was continuously reduced to extract more abstract information and allow translation-

invariance. This was normally achieved by strided convolution or max-pooling. Small objects were found to be difficult to handle as their location information could be lost after down-sampling. To detect the small objects, the feature map has to retain a reasonable resolution, and at the same time being semantically expressive. For this purpose, we applied a sequence of atrous convolutional layers to keep the feature map at the desired resolution. This fits the single-layer representation that detects objects of all scales from one single feature layer, such as the two-stage faster region based convolutional neural networks (R-CNN). The matching qualities of the anchors and ground-truth was found to be highly correlated to the detection performance. A fine feature stride and comprehensive set of anchors helps improve detection performance, especially for the small objects. However, it is a trade-off between detection accuracy and computation cost.

Faster R-CNN [14] is the representative of a "two-stage" object detector, which trained a region proposal network (RPN) to generate object candidates. The candidates were then passed on to another network for multi-class classification and bounding-box fine-tuning. In the second stage, "ROI alignment" [15] ("ROI pooling" in early version) cropped the features for the object proposals and fit them into the same size. "Anchors" were a set of pre-defined bounding-boxes that serve as object proposals for the RPN. The output bounding-box were derived by predicting the "offsets" to the anchors. During training, anchors generate positive and negative examples according to its intersection with the ground-truth.

Skipping the proposal generating process, single shot object detector (SSD) proposed to make final predictions on class label and bounding-box coordinates offsets directly from the feature maps, unlike faster R-CNN that handles objects of all scales on the same feature map. SSD worked on a hierarchical feature pyramid and each feature layer was designated to objects of one scale. YOLO (You Only Look Once) [16] divided the image into a grid. Each grid cell was responsible for predicting the objects whose bounding-box centre lies in this cell. The class label, confidence and bounding-box coordinates were integrated as a single regression problem, which gained processing speed. But one obvious shortcoming is to deal with occluded objects whose centres lie in the same grid cell. Detecting small objects was also found not easy, as the grid division was coarse. In upgraded versions of YOLO [17, 18], anchors were introduced to improve the performance on location prediction. Some recent work proposed to represent the object as coordinate points, and make predictions by grouping the points [19, 20].

A fine feature map resolution is needed to improve detection performance on small objects. Methods for recovering spatial resolution while keeping semantic information were imported from image segmentation as it by nature requires dense prediction on pixel level. A common practice was to use linear up-pooling or transpose convolution (also called "de-convolution") [21] after continuous down-sampling. feature pyramid network (FPN) [22] laterally connected the up-sampled layers to the previous layers to reinforce the information, especially for the shallow layers. This flexible structure could serve as backbone network to many detection schemes. For example, RetinaNet [23] is approximately a combination of SSD and FPN, with a modified loss to mitigate influence of overwhelming numbers of easy negative examples.

In contrast to transpose convolution and up-pooling, atrous convolution (also called dilated convolution or "hole" algorithm) do not down-sample the original image but apply a pyramid of atrous filters with different dilation rates to extract features from different scales. A dilation filter is a normal convolutional filter inserted by zeros. Dilation rate is the distance to insert zeros, which controls the effective receptive field of the filter. This technique was adopted in object detection and applied in numerous occasions, such as road lane detection in [24] and bridge crack detection in [25]. In [26], Atrous convolution was reported to have improved detection performance of small objects. With a different scale of "small" objects, the author used SSD, which we found difficult to match the anchors. And they construct extra layers at the end of the CNN, while we apply atrous convolution in intermediate layers.

A very similar context as our work, African mammals were detected from aerial images in [27], where two sibling networks were constructed. One predicts class probability for each feature map cell, the other outputs bounding-box coordinates. The images were cropped into small pieces and detection was made in each piece. This is a common practice to deal with extremely high-resolution remote sensing images [28] . Obviously, there will be a bunch of objects cut into different parts and this is a problem when preparing training data and stitching the patches back together to form a unified detection.

2. RESEARCH METHOD

2.1. Backbone network

ResNet [29] addressed the degradation problem of very deep neural networks, where accuracy gets saturated when the network goes too deep. By adding the input to the output, very deep network gains detection accuracy. Figure 3 shows the residual unit that performs a convolution operation and "shortcut" connection between the input and output. The idea of residual learning was used in many other architectures like Darknet-53 of YOLO-v3 and Incepton-v3 of GoogleNet [30]. The network could grow very deep by

stacking a bunch of convolutional layers. However, computational cost hampers the use of heavy architectures for big images in our dataset. Herein, we used ResNet-50 that contains 50 convolutional layers.

The main body of ResNet-50 contains 4 blocks that were composed by 3, 4, 6 and 3 layers of the residual units as shown in Figure 3, respectively. Originally, the spatial resolutions of the blocks were 1/4, 1/8, 1/16 and 1/32 of the input image size. Instead of continuously reduce the spatial resolution after Block 2, the output stride was kept at 1/8 of the input image size. To extract abstract features, All the 3x3 convolutional filters in Blocks 2, 3 and 4 are replaced by atrous convolutional filters. Essence of atrous convolution was to catch information from a bigger area and skip some in between by setting "holes" on the filter. The formulation of one-dimensional signals is:

$$y[i] = \sum_{k=1}^{K} x\,[i + r * k]w[k]$$

where $y[i]$ is the output of input $x[i]$ convolved with filter $w\,[k]$ with length k. r is the stride to sample $x[i]$, called "dilation rate". A fine feature map spatial resolution contributes to avoiding omitting the very small objects and makes anchors better matching the ground-truth. But it is a trade-off choice as the number of anchors increase exponentially on a double sized feature map. As the atrous convolutional layers are stacked consecutively and the abstract level accumulates in a very deep network, we propose to use the same dilation rate equals to 2 others than progressively enlarge the dilation rate. Ablation on the dilation rate will be done in Section 5. Bracket part in Figure 4 illustrates the modified ResNet-50 in detail, including layer sizes and depths. The rest depicts the flow of the features derived from the backbone network in Faster R-CNN.

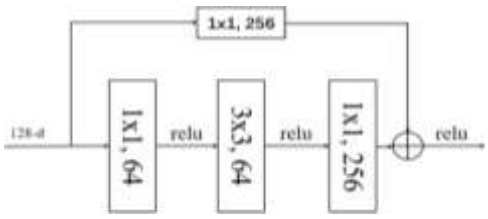


Figure 3. An example of a residual unit of ResNet, the 1x1 filters were used to adjust the feature depths

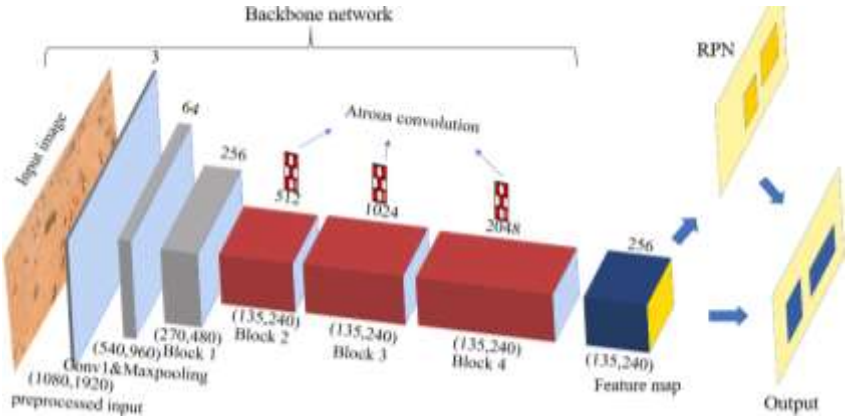


Figure 4. Architecture of the proposed method, the red and white boxes represent the atrous filters that are applied in the backbone network; the numbers on the top show the depth of layers, the numbers at the bottom are the spatial resolution of the layers

The output stride is kept at 8 for several reasons. One is that the smallest object size in our dataset is about 11x11, nominally it would project to at least one-pixel on the feature map at the output stride of 8. Another is that small objects need a small striding step to better match the anchors. The matching quality for each object scale has great impact on the detection results. As the number of small objects compose only small portion of the dataset. Misalignment between the anchors and ground-truth for several examples is not affordable for the small objects. The total number of anchors should also be considered. It is further discussed later together with the detection architecture and anchor settings.

2.2. Detection architecture

The detection architecture follows Faster R-CNN. A 3x3 convolutional filter will slide on the last layer of the backbone network and derives a feature map. Each cell of the feature map outputs predictions for the pre-defined set of anchors. Suppose there are k anchors for each cell, the predictions will be 2k object and non-object scores and 4k coordinate offsets for RPN.

Features of the object proposals generated from the RPN are derived by projecting the bounding-boxes to the feature map. Then through ROI alignment, the features are cropped into size of 14x14 using bi-linear interpolation and further down-sampled to 7x7, as shown in Figure 5. Features of different objects are aligned to the same size that can go through FC layers and output the final predictions. From Figure 1, the most popular object size in our dataset lies in between 48x48 and 64x64, which only projects to 6x6 to 8x8 on the feature map. It is far smaller than 14x14. We experimented with smaller ROI alignment size, such as 7x7 and 2x2 and did not achieve better results. Dividing features of a small objects into fine grained pieces helps describe the object in detail. Intersections of union (IoUs) between the anchors and the ground-truth were used as metrics to select positive and negative examples. For other implementation details of Faster R-CNN, we recommend referring to the original work.

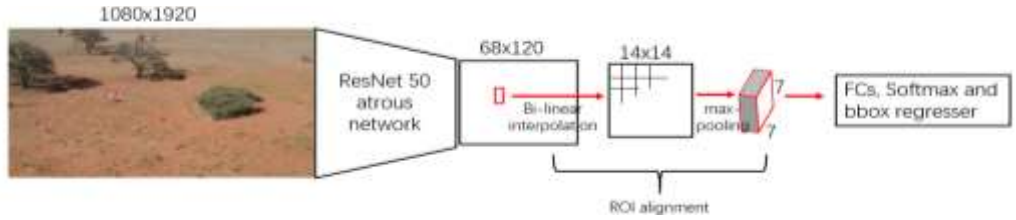


Figure 5. ROI alignment of the second stage; small objects also gain benefits by dividing features to fine grained pieces

2.2. Anchor settings

Anchors were empirically chosen in most of the detection architectures. In Faster R-CNN where predictions were made on a single feature map, the anchors must cover all the object scales. In SSD where predictions were made on multiple feature maps, each feature layer requires a specific designed object scale and overall, the anchors should match all the object scales, in the original work of SSD, the authors used the following formulation to assign the anchors for each feature layer:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), k \in [1, m]$$

m is the number of layers, s_{min} and s_{max} is the minimum and maximum anchor scale that was assigned to the lowest and highest feature layer, respectively. Because the object scale of our dataset is very small and spread in a narrow range, which is reflected in Figure 1 (b), manually choosing the anchors is more flexible. For example, the ratio of a 16x16 anchor box to a 1080x1920 image is 0.00012, making s a very small number. There were semi-automatic anchor assigning methods, such as using centres of K-means clustering on the ground-truth boxes as the anchors. Obviously, K-means clustering only reveals the statistic patterns of the object sizes, but has no clue on the feature stride step, as well as the matching quality between the anchors and ground-truth. Feature stride step is of great importance on the small objects. And carefully balancing the feature map size and matching quality is needed for various detection architectures.

Exhaustive searching of the hyper-parameters such as image size, output stride and anchor scales was implemented for both the single- and multi-layer object detector. The number of anchors and the IoU matching qualities for each object scale are interested factors to be looked upon. Table 1 shows some results for the single-layer detector, we summarize the portion of the biggest IoU that exceeds thresholds 0.5, 0.6 and 0.7 under the combinations of image size, output stride and anchor scales. The aspect ratio is set to [0.5, 1, 2]. Anchor scales are represented by the edge length, for example, anchor scale 8 represents anchor box of size 8x8. For the same image size, the anchor scales seem not have much influence on the IoU matching qualities, probably due to small output stride step and fine division of the anchor scales.

Table 2 shows some of the results for IoU matching qualities for multi-layer detectors. The output stride level is the order of 2 by which the feature map size was reduced. Feature layers of level 2-6 are [2^2, 2^3, 2^4, 2^5. 2^6] times smaller than the input image size. An extra anchor scale in between two anchor scales was added by the choice indicated by "intermediate scale".

Table 1. Anchor settings and matching qualities of single-layer object detector

	Image size	Output stride	Feature map size	Anchor scales	Number of anchors	IoU>0.7	IoU>0.6	IoU>0.5
1	1080x1920	8	(135,240)	[8, 16 ,32, 64, 128]	486000	43.98%	72.78%	99.53%
2	1080x1920	8	(135,240)	[8, 16 ,32, 64, 128, 256]	583200	43.99%	72.81%	99.64%
3	1080x1920	8	(135,240)	[16 ,32, 64, 128]	388800	43.98%	72.78%	99.53%
4	1080x1920	8	(135,240)	[16 ,32, 64, 128, 256]	486000	43.99%	72.81%	99.64%
5	1080x1920	8	(135,240)	[32, 64, 128]	291600	43.91%	72.48%	98.63%
6	1080x1920	8	(135,240)	[32, 64, 128, 256]	388800	43.92%	72.50%	98.73%
7	540x960	8	(68,120)	[8, 16 ,32, 64, 128]	122400	27.63%	60.07%	93.12%
8	540x960	8	(68,120)	[8, 16 ,32, 64, 128, 256]	146880	27.63%	60.07%	93.12%
9	540x960	8	(68,120)	[16 ,32, 64, 128]	97920	27.61%	60.01%	92.83%
10	540x960	8	(68,120)	[16 ,32, 64, 128, 256]	122400	27.61%	60.01%	92.83%
11	540x960	8	(68,120)	[32, 64, 128]	73440	24.80%	49.73%	68.91%
12	540x960	8	(68,120)	[32, 64, 128, 256]	97920	24.80%	49.73%	68.91%

Table 2. Anchor settings and matching qualities of multi-layer object detector

	Image size	Output stride level	Anchor scales	Intermediate scale	Number of anchors	IoU>0.7	IoU>0.6	IoU>0.5
1	1080x1920	2-6	[8, 16 ,32, 64, 128]	N	506430	9.55%	34.30%	78.08%
2	1080x1920	2-6		Y	1012860	28.63%	74.35%	97.85%
3	1080x1920	2-6	[16 ,32, 64, 128, 256]	N	506430	33.39%	70.29%	99.45%
4	1080x1920	2-6		Y	1012860	77.01%	99.90%	100.00%
5	1080x1920	2-6	[32, 64, 128, 256, 512]	N	506430	47.87%	73.28%	98.74%
6	1080x1920	2-6		Y	1012860	92.21%	97.49%	98.75%
7	1080x1920	3-7	[8, 16 ,32, 64, 128]	N	126675	2.31%	8.80%	24.11%
8	1080x1920	3-7		Y	253350	7.43%	24.10%	54.90%
9	1080x1920	3-7	[16 ,32, 64, 128, 256]	N	126675	9.55%	34.31%	78.15%
10	1080x1920	3-7		Y	253350	28.63%	74.34%	97.83%
11	1080x1920	3-7	[32, 64, 128, 256, 512]	N	126675	33.91%	69.71%	98.21%
12	1080x1920	3-7		Y	253350	73.48%	97.09%	98.72%

3. RESULTS AND ANALYSIS

3.1. Dataset

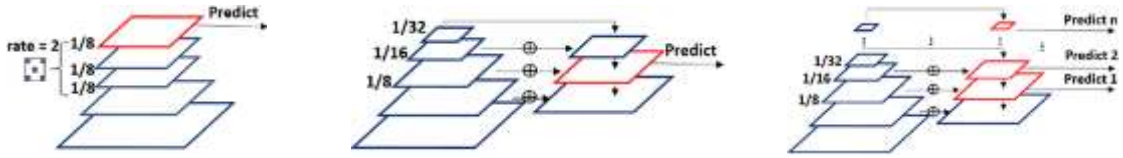
The dataset was collected in semi-desert areas in southern Namibia using DJI phantom 3 and phantom 4 in December (summer in Namibia). We took numerous flights in different times during the day for a week, and the duration of the videos amounted to several hours. Three species were covered in this dataset: blue wildebeest (gnu), gemsbok (oryx), and zebra. These species were chosen because they spread widely in southern Africa, having well representation of the hoofed animals on this land. Additionally, they are gregarious, making it easy to record a number of instances in one shot. Also adding challenges to the detector by introducing crowded and occluded objects.

The resolution of the image is 1080x1920. Frames were taken from the videos every certain interval of seconds and were divided randomly into training and testing sets. 1693 frames with around 20000 instances were used for training, and 389 frames with 4017 instances were used as testing examples. The flight height was 10-20 meters, with sights from various angles to the object. The environment and illumination conditions were diversified on purpose to allow generalization.

3.2. Backbone network

ResNet-50 was taken as the basic feature extracting backbone network for its robustness and moderate size. To retain the output stride at 8, atrous convolutional layers with stride 2 replaced the normal convolutional layers and the stride 2 down-sampling operations. The last layer of the Block 4 was taken as the feature map in the Faster R-CNN detection scheme. Figure 6 (a) is a schematic of the backbone network. The highlighted layers are the activation feature maps. The FPN structure recovers the spatial resolution of the feature map by up-sampling and literal connection as shown in Figure 6 (b). The two network structures could both be used in Faster R-CNN that make predictions on one single feature map.

For multi-layer detectors, directly making predictions from the shallow layers is not feasible. It is better to enhance the sematic information by introducing a FPN structure. SSD takes the output layers of the FPN upon ResNet-50 as shown in Figure 6 (c). YOLO-v3 used anchors to overcome deficiency of location accuracy in earlier versions of YOLO. Predictions were made on a pyramid of 3 feature layers starting from output stride 8. Because of the squeeze-and-excitement structure of the network does similar work as in FPN, no modification was made on the network structure in our experiment.



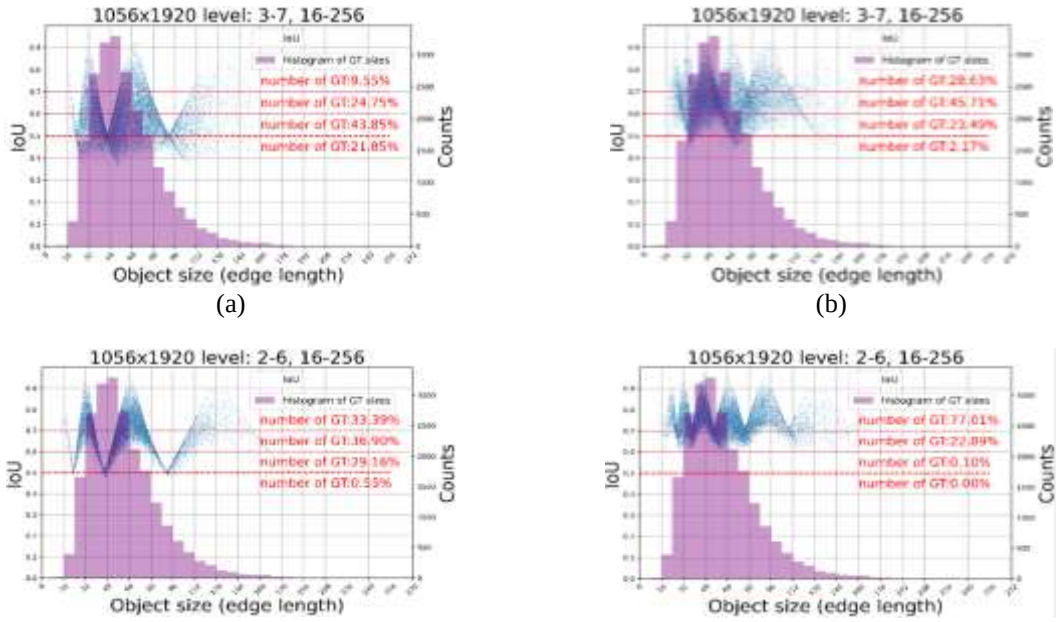
(a) (b) (c)

Figure 6. Activation feature map of various network structures: (a) straight forward ResNet with atrous filters, (b) FPN structure, and (c) SSD style predictions on FPN layers

3.4. Anchors

For small objects in a big images, better matching of the anchors with the ground-truth could only be achieved by elaborate division of anchor scales and a fine resolution feature map. To further reveal the IoU matching for the object scales, Figure 7 plot the biggest IoU of each ground-truth against the object size. The histogram shows the number of objects lie in each object scale. Examples are taken from Table 2, the hyper-parameters are on the top of each figure (Figure 7 (a) and (b)). Intermediate anchors are inserted in the right column compared to the left (Figure 7 (b) to (a) and Figure 7 (d) to (c)). IoU matching is improved by adding intermediate anchor scales, which can be seen by comparing Figure 7 (b) to (a) and (d) to (c). IoU peaks are formed around the new inserted anchor scale. From the top row of Figure 7 to the bottom row ((a) to (c) and Figure 7 (b) to (d)), IoU matching is improved by moving the feature map one layer up, which indicate a finer feature map resolution.

The portion of IoU>0.5 achieves 100% at feature level 2-6 with intermediate anchors as shown in Figure 7 (d). However, the total number of anchors is about 1M, while the number of anchors on the same feature levels without intermediate anchor scales is about 500K. At feature level 3-7 with intermediate anchors, the total number of anchors is only 120K. For single-layer object detectors, the optimum choice is Row 3 of Table 1, the anchor scales are [16, 32, 64, 128] with the original image size. For the multi-layer detectors, Row 4, and Row 6 of Table 2 has comparable IoU matching rate and Row 6 generates less anchors by omitting anchor scale of 16. However, the poorly matched objects are centralized on the small objects. As shown in Figure 8, the objects with biggest IoUs<0.5 are all below 32x32. This influences the detection for small objects. So hyper-parameters in Row 4 is chosen for SSD.



(a) (b) (c) (d)

Figure 7. Visualization of the object size and its best matched IoU (blue dots):
(a) results for feature levels 3-7 and (b) its comparation after intermediate anchors are added, and
(c) feature levels 2-6 and (d) its comparation after intermediate anchors are added

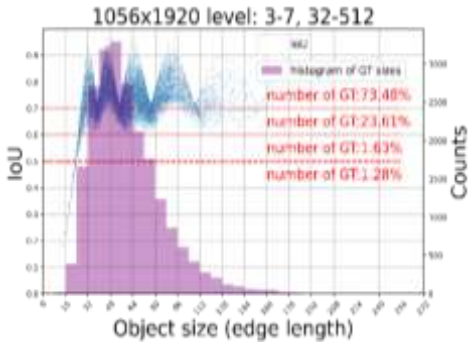


Figure 8. Anchors having high IoU matching rate, but poor matches are centralized on small objects

3.5. Model evaluation

The models were trained from scratch on the training set of our dataset. MS-COCO metrics were used to evaluate the detection performance. Only the overall mAP (mean average precision) and mAP for the large, medium, and small objects were considered, neglecting the corresponding mAR (mean average recall). Table 3 listed the evaluation results for the various detection models, including two-stage detector Faster R-CNN and one-stage SSD and YOLO-v3. The aspect ratios were all set to [0.5, 1, 2]. For Faster R-CNN, the backbone networks were based on ResNet-50. Hyper-parameters HP1 and HP2 defined the image size and anchor scales. HP1 was the settings with the optimum anchor choice balancing the IoU matching rate and the total number of anchors. Under the same backbone network with atrous filters. HP1 outperforms HP2 as the IoU matching rate between the anchors and ground-truth is higher. Rate1 and rate 2 defined the different dilation rates. Using the same smallest dilation rate [2, 2, 2] led to better results than enlarging the dilation rate for each block of Renet-50. This could be caused by the narrow range of the object scale combined with elaborate division of the anchors that makes enlarging the "receptive field" not necessary. HP2 uses image size that is half of the original image, which leads to much lighter network. But the performance is inferior due to the worse anchor matching rate and loss of information.

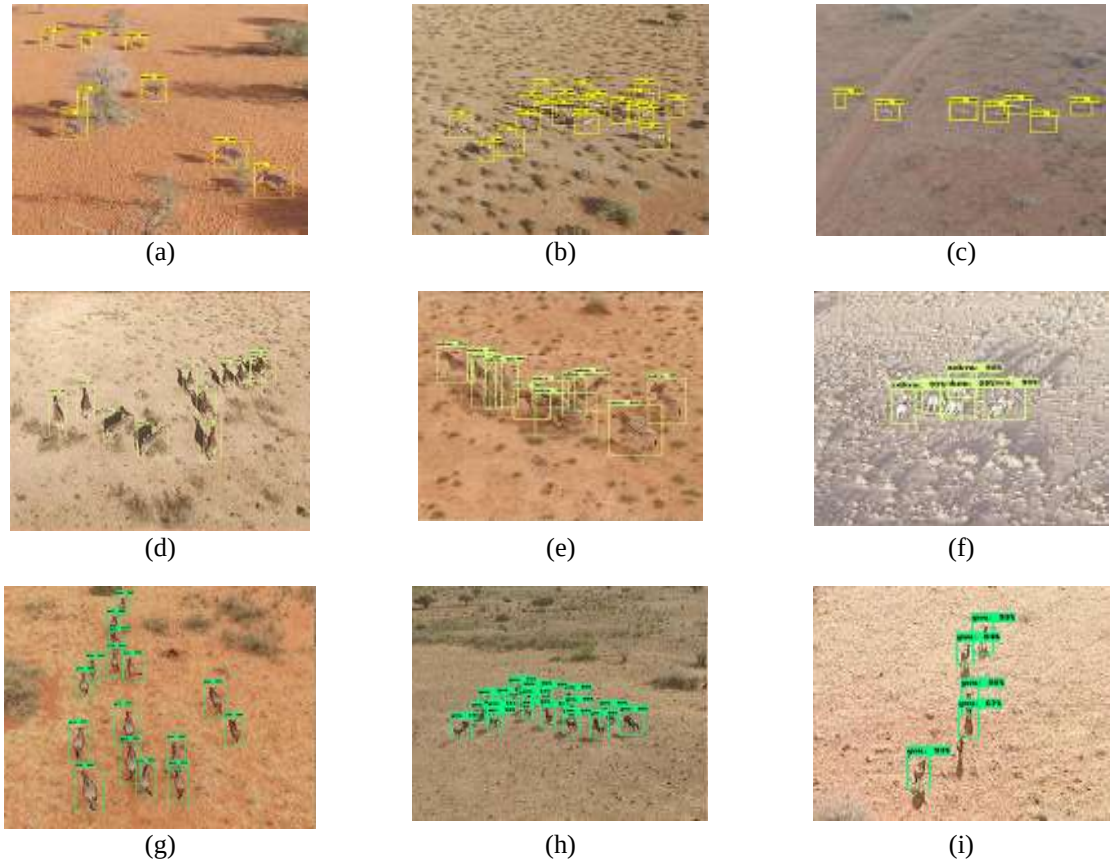
Table 3. Evaluation of various models

Detection model	Backbone	Hyper-parameters	mAP	mAP (large)	mAP (medium)	mAP (small)
	ResNet-50+atrous	HP1+rate1	0.59	0.69	0.6	0.36
Faster	ResNet-50+atrous	HP1+rate2	0.42	0.62	0.58	0.33
R-CNN	ResNet-50+atrous	HP2+rate1	0.51	0.67	0.52	0.23
	ResNet-50+FPN	HP1	0.48	0.64	0.49	0.17
SSD	ResNet-50+FPN	SHP	0.32	0.34	0.33	0.28
YOLO-v3	Darknet-52	YHP	0.34	0.52	.0.36	0.31
HP1: image size 1080x1920, anchor scales: [16,32,64,128]						
HP2: image size 540x960, anchor scales: [8,16,32,64]						
Rate1: dilation rate [2,2,2]						
Rate2: dilation rate [2,4,8]						
SHP1: image size: 1080x1920, feature level: 3-7, anchor scales: [16,32,64,128,256], intermediate scale: Yes						
YOLO-v3: image size: 1080x1920, anchor scales [16, 64, 256]						

SSD could only take the feature layers from a FPN structure. SHP, the hyper-parameters for SSD was the optimum option that was analysed previously. The detection performance was inferior for all object scales than the two-stage Faster R-CNN. Fine division of the anchor scales was compulsory for such data that were both objectively and relatively small. However, designing a proper anchor set for a sequence of feature layers is not easy. Good IoU matching for the object scales on both the small and the large ends is difficult to be met at the same time. YOLO-v3 outputs three feature layers, the anchor scales are set to [16, 64, 256]. Performance improvement was gained on all object scales over SSD with FPN structure.

Figure 9 gives visualizations of some good detection results. Figure 9 (a)-(c), (d)-(f), and (g)-(j) are results for the species: zebra, oryx and blue wildebeest, respectively. The scenario is where the animals are clear and parted, cluttered, and occluded by one another, in distance and small in size and the corresponding results are Figure 9 (a), (d), and (g); Figure 9 (b), (e), and (h); and Figure 9 (c), (f), and (i); respectively. The pictures are cropped and resized for display convenience and do not reveal their real size of the full image. The objects that are densely cluttered and small in size are notoriously difficult to detect, our method shows good results in some places.

Examples of false detections are shown in Figure 10. Figure 10 (a)-(c) are false positives for zebras, oryx and blue wildebeest, Figure 10 (d)-(f) are the animals not being detected (false negatives). The false negatives are mainly the small objects. Adding more training examples of the small objects in the future may help the situation.

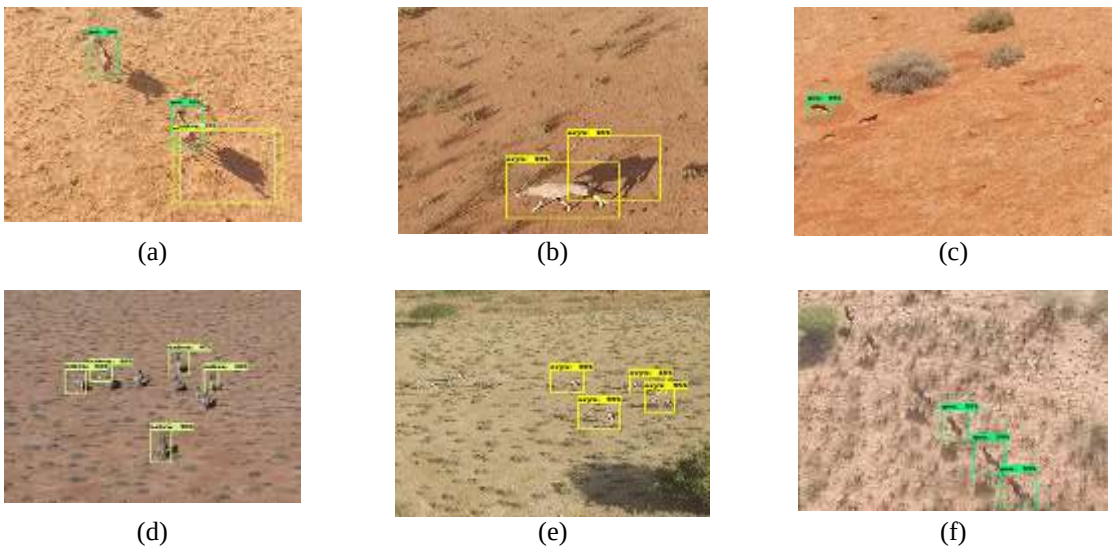


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 9. Detection results of (a) oryx when they are apart, (b) crowded and occluded, and (c) for the small objects; (d) zebra when they are apart, (e) crowded and occluded, and (f) for the small objects; and (g) blue wildebeest when they are apart, (h) crowded and occluded, and (i) for the small objects



(a) (b) (c)

(d) (e) (f)

Figure 10. False positive detections for (a) zebra, (b) oryx, and (c) blue wildebeest; and false negative detections for (d) zebra, (e), oryx, and (c) blue wildebeest

4. CONCLUSION

We brought up the problem of detecting African hoofed mammals using aerial imagery taken from UAVs. In a ResNet-50 backbone network, a sequence of atrous convolutional filters were used to keep the feature map resolution at certain level and at the same time continue extracting deeper abstract semantic features. The detection performance for such data is sensitive to the matching quality between the anchor and the ground-truth. The image size, output stride and anchor choices combined together determine the IoU matching qualities. This feature extraction technique was proven robust in detecting small objects by comparing to FPN. The two-stage Faster R-CNN surpass single-stage detectors in detecting small objects, especially for our dataset where the object size is of small ratio to the full image.

ACKNOWLEDGEMENTS

The videos used in this paper were collected under the assistance of Gondwana Collection, Namibia.

REFERENCES

[1] B. Kellenberger, M. Volpi, and D. Tuia, "Fast animal detection in UAV images using convolutional neural networks," *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, TX, USA, 2017, pp. 866-869.
[2] C. Chen and K. Liu, "Stingray detection of aerial images with region-based convolution neural network," *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, Taipei, Taiwan, 2017, pp. 175-176.
[3] L. B. Boudaoud, F. Maussang, R. Garello, and A. Chevallier, "Marine Bird Detection Based on Deep Learning using High-Resolution Aerial Images," *OCEANS 2019 - Marseille*, Marseille, France, 2019, pp. 1-7.
[4] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," *Sensors*, vol. 19, no. 7, pp. 1-16, 2019.
[5] G.-S. Xia, *et al*., "DOTA: A large-scale dataset for object detection in aerial images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3974-3983.
[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303-338, 2010.
[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. F. Li, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248-255.
[8] Y. Su and C.-C. J. Kuo, "On extended long short-term memory and dependent bidirectional recurrent neural network," *Neurocomputing*, vol. 356, pp. 151-161, 2019.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.
[10] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," in *IEEE Computational Intelligence Magazin*e, vol. 13, no. 3, pp. 55-75, Aug. 2018.
[11] Y. Su, K. Fan, N. Bach, C.-C. J. Kuo, and F. Huang, "Unsupervised multi-modal neural machine translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10482-10491.
[12] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 2013, pp. 8614-8618.
[13] H.-C. Shin, *et al*., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, May 2016.
[14] 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, 2016.
[15] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.
[16] 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.
[17] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.
[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[19] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734-750.
[20] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569-6578.
[21] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 1 April 2018.
[22] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.
[24] Y. Sun, L. Wang, Y. Chen, and M. Liu, "Accurate Lane Detection with Atrous Convolution and Spatial Pyramid Pooling for Autonomous Driving," *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dali, China, 2019, pp. 642-647.
[25] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, "Automatic bridge crack detection using a convolutional neural network," *Applied Sciences*, vol. 9, no. 14, pp. 1-14, 2019.
[26] T. Guan and H. Zhu, "Atrous Faster R-CNN for Small Scale Object Detection," *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, Wuhan, China, 2017, pp. 16-21.
[27] G. K. Verma and P. Gupta, "Wild animal detection using deep convolutional neural network," in *Proceedings of 2nd international conference on computer vision & image processing*, Springer, 2018, pp. 327-338.
[28] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, 2012, pp. 4379-4382.
[29] 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.
[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.