Animal Detection in Highly Cluttered Natural Scenes by using Faster R-CNN

Wenjun Yu, Sumi Kim, Jeong-Hyu Lee, Jaeho Choi

Abstract— With the increasing awareness of environmental protection, people are paying more and more attention to the protection of wild animals. Their survive-al is closely related to human beings. As progress in target detection has achieved unprecedented success in computer vision, we can more easily tar-get animals. Animal detection based on computer vision is an important branch of object recognition, which is applied to intelligent monitoring, smart driving, and environmental protection. At present, many animal detection methods have been proposed. However, animal detection is still a challenge due to the complexity of the background, the diversity of animal pos-es, and the obstruction of objects. An accurate algorithm is needed. In this paper, the fast Region-based Convolutional Neural Network (Faster R-CNN) is used. The proposed method was tested using the CAMERA_TRAP DATASET. The results show that the proposed animal detection method based on Faster R-CNN performs better in terms of detection accuracy when its performance is compared to conventional schemes.

Index Terms: deep learning, convolutional neural networks, faster r-cnn, animal recogni-tion, RPN

I. INTRODUCTION

Due to the rapid growth of the human population and the relentless pursuit of eco-nomic development, the earth's ecological system is going through tremendous changes and exposed to irreversible vicious cycles. Human activities have changed the surface area of the land, population, habitats, and behavior of wildlife. What's more serious is that many wild species on Earth are in the verge of extinction. No longer, people can find their traces, or many of them are taken to new areas where they might disrupt the ecological environment and even destroy the natural and hu-man systems. Therefore, monitoring wildlife is critical because it can provide re-searchers with valuable information to come up with means for conservation and management of diverse, balanced and sustainable ecosystem.

In the intelligent video analysis [1,2], detecting and segmenting moving objects from the background is an important and feasible step. In the past two decades, a large number of studies have conducted background modeling and foreground object detection [3-5]. However, methods that are robust and versatile enough to handle complex natural dynamic scenes are still very limited. Videos taken in the natural environment represent a large number of challenging scenes that are not fully dis-cussed in the literature. These types of scenes are often highly chaotic with swaying trees in the wind, water rippling, moving shadows, sunspots, and rain, etc. It be-comes more complex when natural animal camouflage adds additional complexity to the analysis. The key challenge of these scenarios is how to establish an effective model to capture the complex background motion and structural dynamics while maintaining enough discrimination to detect and segment the foreground of the ani-mal. Traditional mobile-based technologies are not suitable here because the back-ground is highly dynamic.

Recently, approaches based on deep neural networks, such as R-CNN [6] and var-iants Fast-R-CNN [7], are achieving the state-of-the-art performance in object detec-tion. Normally, these methods have two main components: The first component is object region proposal. It is used to scan the entire image to generate a set of candi-date image regions or bounding boxes at different locations and scales that could possibly contain the target objects; the second one is image classification. This de-termines whether the proposed areas are truly objects or not.

In this paper, according to Faster R-CNN [8], a novel animal detection method is put forward. The proposed system is trained on the CAMERA_TRAP DATASET [9]. The experiments are performed to verify the effectiveness of the system detecting the animals. The structure of this paper is organized as follows. Section II describes the proposed animal detection method. Section III displays the experimental process-es and results. Finally, Section IV summarizes the whole paper.

II. PROPOSED ANIMAL DETECTION METHOD

The Faster R-CNN is originally a work done by Kaming. It is also a deep learning framework that is well-known and widely used in the field of target detection. Since it was proposed in 2016, it has become a baseline of the detection field. The Faster R-CNN realizes a stage-to-stage detection process of target detection in the R-CNNs series. The entire R-CNNs series framework consists of three parts: (1) proposal forwarding; (2) CNN for extracting the features of the proposal area; (3) identification and regression of the location area for completing the inspection process. On the other hand, the Faster R-CNN framework consists of two parts: (1) a Region Proposal Network (RPN) for forwarding proposal; (2) a detector for classification and regression. The Faster R-CNN framework is shown in Fig 1. RPN extracts the

Revised Version Manuscript Received on August 19, 2019.

Wenjun Yu, Department of Electronic Engineering, CAIIT, Chonbuk National University, Chonju, Korea 561-765.
Sumi Kim, Seoyeong University, Gwangju, Korea.
Jeong-Hyu Lee, His Department Name, University/ College/ Organization Name, Chonju, Korea.
Jaeho Choi, Dept. of SW Engineering, Chonbuk National University, Chonju, Korea.

features of the entire image, and the proposal is put forward. The proposal feature map area is placed in the detector for classification and regression.

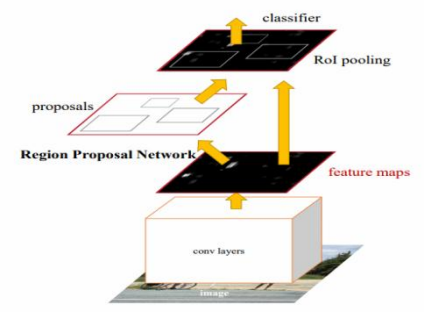

Fig. 1. Fast R-CNN framework

Region Proposals Network

The Faster R-CNN is combined with the RPN network [8] and the Fast R-CNN network [8]. As shown in Fig.2, the red dashed box is the RPN and the blue dashed box is the Fast R-CNN. The proposal obtained by RPN is directly connected to the ROI pooling layer. So, the Faster R-CNN is a framework for CNN network to achieve end-to-end object detection, with good robustness. The main function of the RPN is to convert an input image into a number of rectangular boxes; each box has a corresponding object score. In fact, the RPN network is very simple. It adds an n*n (n=3) convolution layer to the existing public convolution layer; on the basis of this convolution layer it has two 1*1 sibling layers, i.e., classification layer and regression layer.

The Anchor mechanism is the most important part of the entire Faster R-CNN. When the convolution layer is convolving, the filter window translates in steps of the stride length on the output of the upper layer, and then calculates. Such a translation method coincides with the sliding window technology applied in the early target detection field. Thus, we can think of the n∗n convolution layer as a n∗n window sliding on the input feature map. Then we assume that there might be k targets for each sliding window, then 4k need to output (corresponding to the four points of the rectangle) for the behind regression layer, and 2k outputs for classifying layer (corresponding to the probability of each target is the target or not). The box corresponding to the k targets are called the anchor. Determining anchor is actually a reverse process of ROI Pooling. If it knows the exact target area in ROI Pooling, then it can divide the target area into several blocks for pooling. By dividing the target area into 3∗3, the final Pooling result is a 3∗3 square.

In order to train the RPN network, a multitasking loss is designed: one for representing the classification of an object; the other for indicating positional regression. Since there are 9 anchors in all positions of a feature map, many anchors would be generated. During the training process, the generated anchors need to be tagged for supervision and training. The author regards the intersection-over-union (IoU) of arbitrary ground truth greater than 0.7 as a positive sample; it considers any ground truth IoU less than 0.3 as a negative sample. The samples with IoU between 0.3 and 0.7 do not participate in training. The multitasking loss function is defined as follows:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \varphi \frac{1}{N_{cls}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

In the formula i is the anchor in the mini batch; p_i is the probability of the object i anchor; p_i^* is the ground truth; t_i is the parameter corresponding to the four corners of the predicted rectangle; t_i^* is for the ground truth; L_{cls} is the log loss values of the two types, $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$; R is the smoothness of L_1 function in Fast R-CNN. The meaning of p_i^* is that only the rectangle of the object, i.e. , $p_i^* = 1$ would participate in the loss calculation. Using the Faster R-CNN, we only need one time to process the input image in the whole computing processes. Then, improve the bound boxes of region proposals. Because of convolutional layers can share each other, it can easy to improve the performance of object detection that uses a very deep convolution network to train.

III. EXPERIMENTAL SETUP AND RESULTS

Datasets

In this work, the camera trap dataset is used for performance evaluation. It provided more than 1 million images for animals. In fact, these images have 23 kinds of animals with two formats including daytime color and nighttime infrared.

Model Training

Normally the number of positive samples is less than the number of negative samples in the anchor; 256 samples made of 128 positive samples and 128 negative samples were selected at random from images when training the RPG network. We train our model using a 16x16 patch and a codebook size of 128. The region candidates were regarded as the true positive patches when its IoU with the ground truth is greater than 0.5. The others were treated as the false positives. We chose IoU = 0 patches as the negatives in order to avoid mixing possible animal object patches, in practice.

Results

Table 1 summarizes the results of the Faster R-CNN and other methods. We can see that the Faster R-CNN average is higher than others and each item is also higher than the others. It results experimentally verify that the Faster R-CNN can detect the animals better than other methods.

Table I. Performance comparison on Camera-trap-dataset

	EC Best	YOLO[10]	Fast-RCNN[7]	Faster-RCNN[8]
Train set Fine tune set		voc07+ voc12 camera-trap	voc07+ voc12 camera-trap	voc07+ voc12 camera-trap
Agouti	0.7632	0.7593	0.742	0.7514
Collared Peccary	0.8209	0.8359	0.8015	0.8094
Pace	0.7969	0.8169	0.8039	0.8289
Red Squirrel	0.8563	0.8915	0.8517	0.8879
White-nosed Coati	0.8059	0.8314	0.7899	0.7952
Spiny Rat	0.7539	0.7642	0.7193	0.7314
Ocelot	0.7918	0.8192	0.7726	0.7952
Red squirrel	0.7345	0.7682	0.7328	0.7437
Common Opossum	0.7816	0.8164	0.7951	0.8155
Bird spec	0.6527	0.7465	0.6412	0.6619
Great Tinamou	0.789	0.8349	0.8035	0.8148
White-tailed Deer	0.8218	0.8432	0.8303	0.8792
Mouflon	0.7594	0.8448	0.7692	0.7846
Red Deer	0.7947	0.8214	0.7963	0.7991
Roe Deer	0.7969	0.8391	0.7793	0.7925
Wild Boar	0.7863	0.8417	0.7965	0.805
Red Fox	0.6471	0.7349	0.6752	0.6849
European Hare	0.7156	0.7514	0.7391	0.7485
Wood Mouse	0.7094	0.7539	0.7293	0.7336
Coiban Agouti	0.7316	0.7815	0.749	0.7598
Average	0.7824	0.8315	0.7801	0.7886

IV. SUMMARY

In this paper, we presented a deep learning method for animal detection using the Faster R-CNN. The architecture of the Faster R-CNN was described along with procedures for system training. The proposed system is evaluated, and its performance is compared to the conventional method. The simulation results using well-known data set have shown us that the Faster R-CNN has a robust performance detecting animals presented in challenging images with highly cluttered and dynamic natural scenes. It outperforms other methods such as R-CNN, YOLO and Fast R-CNN.

V. ACKNOWLEDGEMENTS

This work was supported partly by funds provided by BK21+ and Chonbuk National University of Korea.

REFERENCES

1. Y. Sheikh and M. Shah, “Bayesian modeling of dynamic scenes for object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 11, pp. 1778–1792, 2005.
2. T. Ko, S. Soatto, and D. Estrin, “Background subtraction on distributions,” Proc. 10th Eur. Conf. Comput. Vis., pp. 276–289, 2008.
3. Y. Ren, C.-S. Chua, and Y.-K. Ho, “Motion detection with non-stationary background,” Mach. Vis. Appl., vol. 13, no. 5, pp. 332–343, 2003.
4. C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” ACM Trans. Graph., vol. 23, pp. 309–314, 2004.
5. Y. Boykov and V. Kolmogorov, “An experimental comparison of mincut max-flow algorithms for energy minimization in vision,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, pp. 1124–1137, 2004.
6. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comp. Vis. Pattern Rec., pp. 580–587, 2014.
7. R. Girshick, “Fast r-CNN,” in Proc. Int. Conf. Comp. Vis., pp. 1440– 1448, 2015.
8. K. H. Shaoqing Ren and J. S. Ross Girshick, “Faster R-CNN: Towards real-time object detection with region proposal networks,” Adv. Neural Inf. Process. Syst., pp. 91–99, 2015.
9. videonet.ece.missouri.edu/ cameratrap/videonet.ece.missouri.edu/cameratrap/.
10. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” CoRR, 2015. (http://arxiv.org/abs/1506.02640)