Wild animals detection based on YOLOv5

Xudong Li

School of Mathematics, Taiyuan University of Technology, Jingzhong, Shanxi province, 030024, China

lixudong5044@link.tyut.edu.cn

Abstract. Wild animal monitoring is of great significance to population discovery and research on animal behavior and habits. Early wild animal monitoring mainly relied on human effort, which is time-consuming and contains safety risks. In recent years, with the continuous development of pattern recognition techniques, automated wildlife detection algorithms based on image content analysis have also made progress thanks to these achievements. However, due to the complexity of field scenes, the recognition accuracy and robustness of existing methods can not meet the practical application requirements. Based on considerations, we suggest a field animal detection method based on YOLOv5, which aims to localize and recognize the wild animal. We analyzed the change of recognition accuracy for different scenes in detail, especially for the scene containing multiple targets, small targets or occluded targets. We have used a large number of experiments to verify the feasibility of this method.

Keyword: animal detection, deep learning, YOLOv5.

1. Introduction

The identification and tracking of wild animals is crucial for analyzing the population size and studying their behavior habits of specific organisms, which is one of the basic tasks of biologists in field investigations. Early detection of wild animals mainly relies on the observation of human eyes, which will cost a lot of manpower and time. Due to the gradual popularity of portable image acquisition devices, people begin to achieve animal recognition by analyzing the image information. However, it is very difficult for people to quickly distinguish the animals appearing in the picture, especially when they are in a strange environment. Typically, photographers or scientific researchers hire local guides to guide them during their first visit. However, the assistance that guides can provide is still limited, and they may not be well suited to the work needs of photographers or scientific researchers, and some people may not hire guides. At the same time, it is often difficult for people to cope with the large amount of image information quickly provided by modern devices, such as conducting wide-area search on animals through drones. In this case, it is undoubtedly costly and inefficient for people to personally identify the animals in the image. Therefore, how to quickly identify wild animals is very important for people who need outdoor exploration.

Thanks to the rapid development of artificial intelligence technology, animal detection based on pattern recognition has gradually matured. Early field animal detection methods mostly relied on manual features, where biologists design features that can efficiently and accurately express the characteristics of animal categories by considering the characteristics of animals, such as shape, contour, color, and so on. Using these manually designed features, a detector is further introduced to achieve detection. As one of the representatives of traditional detection algorithms, Viola-Jones uses Haar-like features to quantify differences between pixel totals and local image features in different regions, which can capture the contour and appearance of an object. However, due to color difference recognition, the color of the object itself can have an impact on detection. In addition, the HOG algorithm is also very famous. The main idea of HOG algorithm is to first convert a image from color to a single gray, and then obtain the gradient direction histogram of the image. The gradient direction histogram can well reflect the edge information of the detected target, thereby obtaining the approximate contour and shape of the target. DPM algorithm has achieved the best result in traditional detection algorithms. The DMP algorithm is a component based detection algorithm that uses ideas such as gradient direction histograms, sliding windows, and treats the detected object as a combination of several components. However, due to the addition of many heuristic rules to DPM, the training process is relatively complex [1].

The above methods have to some extent freed scientific researchers from the onerous task of image analysis. However, limited by the expression capabilities of manual feature, the recognition accuracy of the above methods still do not reach the practical level. In recent years, many people have made contributions to the development of deep learning, and visual detection algorithms have benefited from this, making continuous breakthroughs in the accuracy and speed of wildlife detection [1]. According to the differences in algorithm design ideas, there are roughly two types of existing target algorithms based on deep learning: two-stage detection and one-stage detection. First one mainly contains two processes: "selecting acceptable candidate boxes" and "using CNN for classification and recognition". The R-CNN algorithm is the first algorithm model to introduce CNN into the field of target detection, and many subsequent models have been improved from R-CNN, including the well-known Fast RCNN [2]. The Feature Pyramid Network is also a representative two-stage detection. The one-stage detection only requires the image to be fed into the neural network once to predict the boundary box where the target is located. Generally speaking, two-stage detection has higher accuracy, while one-stage detection is faster, enabling better real-time detection [1, 3], and is more suitable for application on edge mobile devices.

In this paper, based on the performance requirements for detection of wild animals, the YOLOv5 algorithm belongs to one-stage detection is selected to achieve the target detection of wild animals. After several generations of improvement, YOLO algorithm has introduced many mainstream ideas and methods, achieving a satisfactory balance between its running speed and accuracy [4, 5], making it relatively easier to carry on different devices. Based on the model construction, we analyzed the animal detection accuracy of YOLOv5 in different field scenarios in detail to verify the feasibility of this method.

2. Method

2.1. Revisiting YOLOv5 algorithm

The characteristic of the YOLO algorithm is that it can maintain a certain accuracy rate at the same time with high speed, and has a relatively low probability of treating the background as an object to be recognized[6]. It can present good classification results for various objects and has strong versatility.

The most basic idea used by the YOLO series algorithm is to divide a picture into several grids of the same size [6]. Those grid will be responsible for predicting one target which falls into the grid, and each grid contains several parameters, including the location (x and y) of the object to be detected, the shape (length and width) of the candidate box, the probability level, and the type of object to be detected. Its neural network will output these parameters as the target and extract features around this target. The subsequent versions of YOLO is based on the original YOLO, actively adopting popular and useful new ideas and methods [7, 8], and finally becoming the YOLOv5 model with stable performance.

2.2. Network structure

The network structure adopted by YOLOv5 contains three different parts: Backbone, Neck and Head. The first part uses CSP-Darknet53 [3] and replaces SPP with SPPF to improve performance. Neck uses PANet to improve network capability [5, 9]. In the Head, YOLOv5 uses the same three detection layers used in previous version.

The following is the Backbone details of YOLOv5s, which can be observed in Table 1. Here, “from” refers to the layer where to obtain parameters, and the value of “-1” in there means the previous layer. “Number” refers to the number of modules that will be used. If the number is not 1, its size will be subjected to depth_multiple. The parameters “depth_multiple” and “width_multiple” will be mentioned later. “Module” refers to the type of module used in that layer. The details are defined in the common.py file. Among them, “Conv ” is CBS, including a Conv2d, a BatchNorm2d and a SiLU. “C3 ” is CSP, including three CBS and one Bottleneck in there. It should be noted that C3 is different in neck and here. Furthermore, YOLOv5 uses SPPF to replace the previous SPP, which significantly improved the speed of calculation process. “Args” refers to the output parameters of the module, which also affects the number of inputs in the following layers. Its size will be subjected to width_multiple.

Table 1. Backbone structure of YOLOv5s.

from	number	module	args
-1	1	Conv	[64, 6, 2]
-1	1	Conv	[128, 3, 2]
-1	2	C3	[128]
-1	1	Conv	[256, 3, 2]
-1	6	C3	[256]
-1	1	Conv	[512, 3, 2]
-1	9	C3	[512]
-1	1	Conv	[1024, 3, 2]
-1	3	C3	[1024]
-1	1	SPPF	[1024, 5]

As shown in Table 2, the first four blocks are actually the Neck part of YOLOv5 and the last block is Head. In this model, “Conv” is still CBS. “nn.Upsample” is a built-in upsampling module of Pytorch. “Concat ” is used to splice the output results of different two-layer modules. It is worth noting that C3 in Neck only contains three CBS, and Bottleneck is not used. The Neck part of YOLOv5 adopts both top-down and bottom-up feature extraction methods, and combines the results in different scales to obtain more accurate results.

Table 2. Network structure of Neck and Head in YOLOv5s.

from	number	module	args
-1	1	Conv	[512, 1, 1]
-1	1	nn.Upsample	[None, 2, 'nearest']
[-1, 6]	1	Concat	[1]
-1	3	C3	[512, False]
-1	1	Conv	[256, 1, 1]
-1	1	nn.Upsample	[None, 2, 'nearest']
[-1, 4]	1	Concat	[1]
-1	3	C3	[256, False]
-1	1	Conv	[256, 3, 2]
[-1, 14]	1	Concat	[1]
-1	3	C3	[512, False]
-1	1	Conv	[512, 3, 2]
[-1, 10]	1	Concat	[1]
-1	3	C3	[1024, False]
[17, 20, 23]	1	Detect	[nc, anchors]

The Head part accepts the characteristic diagrams of three different scales. Because the target detection algorithm is generally difficult to detect small targets, the corresponding feature map with larger scale has greater weight in calculation. In YOLOv5 , there are nine candidate boxes, three under each scale, which are used to predict objects of different sizes [3, 9]. The specific size of candidate boxes is provided in yolov5s. yaml.

YOLOv5 also allows two grids around the grid where the center of the detected target is located to participate in the prediction process to increase the number of positive samples. At the same time, the predicted values of position is allowed to fluctuate between -0.5-1.5. In YOLOv5 , the matching method of the candidate box has also changed. When the aspect ratio of the bounding box to the ground truth is within 4 times, the bounding box will be selected. In this way, more positive samples can also be obtained for training. So YOLOv5 allows the predicted values of w and h to fluctuate between 0-4.

2.3. *Loss function*

The loss function adopted by YOLOv5 contains classes loss, objectness loss and localization loss.

In the part of classes loss, BCE loss is adopted to allow the prediction result to be multi-label [7]. In the following expression, N is the number of classes that can be considered, $\mathrm{y_i}$ is the estimate of the corresponding class, and $\mathrm{y_i^*}$ is the true value of the corresponding class.

$$\mathrm{L_{class}} =-\sum_{n=1}^{N} y_i^* * \log(\mathrm{Sigmoid}(y_i)) + (1-y_i^*) * \log(1-\mathrm{Sigmoid}(y_i)) \quad (1)$$

In the Objectness loss formula, gr is whether the target exists. If it exists, it is 1. Otherwise, it is 0. score_iou is the CIoU of the bounding box and the ground truth.

$$\mathrm{L_{obj}} = (1 - \mathrm{gr}) + \mathrm{gr} * \mathrm{score\_iou} \tag{2}$$

Localization loss can be expressed as:

$$\mathrm{L_{local}} = 1 - \mathrm{CIoU(B, B_{gt})} \tag{3}$$

$$\mathrm{CIoU(B, B_{gt})} = \mathrm{IoU(B, B_{gt})} - \frac{\rho^2\mathrm{(B,B_{gt})}}{\mathrm{c}^2\mathrm{(B,B_{gt})}} - \frac{\mathrm{v}^2}{1-\mathrm{IoU(B,B_{gt})}+\mathrm{v}} \tag{4}$$

$$\mathrm{IoU(B, B_{gt})} = \frac{|\mathrm{B} \cap \mathrm{B_{gt}}|}{|\mathrm{B} \cup \mathrm{B_{gt}}|} \tag{5}$$

$$\mathrm{v} = \frac{4}{\pi} * (\arctan\frac{\mathrm{w_{gt}}}{\mathrm{h_{gt}}} + \arctan\frac{\mathrm{w}}{\mathrm{h}})^2 \tag{6}$$

Where $\rho(\mathrm{B, B_{gt}})$ is the distance between the bounding box and the ground truth, and $\mathrm{c(B, B_{gt})}$ is the diagonal length of the rectangle contains the bounding box and the ground truth.

When YOLOv5 detects multiple candidate boxes and the candidate boxes overlap, YOLOv5 will use NMS to select the most suitable candidate box. NMS will first sort the overlapping candidate boxes according to their confidence, then select several suitable candidate boxes by calculating the weight, and finally select the best one.

In order to get more samples for training and improve the effect of the model, YOLOv5 adopted a variety of data enhancement methods, including Mosaic [8], Copy Paste, Random affine, etc. At the same time, YOLOv5 adopts a variety of training strategies, including Multi-scale training, AutoAnchor, etc. These strategies play a great role in training your own training set.

2.4. Model training

In the actual training process, the data set for training needs to be prepared. First, we need to select a portion of the data set as the training set and the remaining portion as the verification set, then put them into the training folder and verification folder under the images folder respectively. In addition, it is also necessary to give appropriate YOLO format labels to the positive samples for each pictures. Each label has five parameters of type, x, y, w, and h relative to the size of the picture. It is saved as a txt file with the same name as the picture, and also placed in the train folder and verification folder under the labels folder.

This time, the data set is taken from African Wildlife in Kaggle, which includes four animals: buffalo, elephant, zebra and rhino. Each animal has nearly 400 pictures. Here, We selected 300 pictures from each animal in data set for training, and other pictures will be used for verification.

In addition, three documents need to be prepared in advance. The first is the weight file of YOLOv5 . Here, yolov5s.pt is used. It has a relatively small size, but it can ensure good accuracy, which is very suitable for this case. Next, YOLOv5 needs a yaml file for the samples. The file contents include the path of the training and verification data set, the total number and names of possible classes. Furthermore, the parameter in yolov5s.yaml also needs to be changed. The parameter “nc” should be set as the total number of possible classes.

In the train.py file used for training, several parameters need to be changed: set “weights ” as the path of weight file , set “cfg ” as the path of yolov5s.yaml file, set “data ” as the path of the data set yaml file, set “epochs” to 20, set “batch-size” to 4, and set “workers ” to 4. The “batch-size” and “workers ” are set to 4 because of the limitation of the equipment used during training.

3. Training process and performance analysis

3.1. Original data

The original data comes from the data set African Wildlife in Kaggle. The original file contains four folders, under which are placed photos of animals with corresponding folder names. These animal photos have a three-digit number as their name, and there are also txt files with the same name under the folders. These txt files contain YOLO format labels for the corresponding photos, indicating the classes of positive sample appearing in the images, the location, width and length of the prediction box. All of the information for one label will be placed on the same line. The data set includes four animals: buffalo, elephant, rhino, and zebra. Each animal has 376 corresponding images and label files. These images may contain multiple and diverse animals, which increases the diversity of samples and allows neural network to produce better results.

3.2. Evaluation metric

A variety of indicators used to evaluate the model will be shown below. These are commonly used indicators to intuitively evaluate the results.

There are three kinds of broken lines formed by the loss function used in YOLO during the training process. Each training will generate a node, and there will be a corresponding broken line for the boundary box, confidence and class of target. A decrease in the broken line means that the result gradually converges, and the neural network achieves higher accuracy. Each broken line is further divided into two types: train and val, which represent different values of the loss function in the training set and the verification set. Generally, as the number of training time increases, the broken lines in the training set will decline relatively steadily, while the broken lines in the verification set is prone to fluctuations.

Here are also broken lines and confusion matrix that display neural network precision, recall, and mAP. Each training will also generate a node. These indicators can intuitively display the accuracy of neural network results. The higher the broken line, the higher the accuracy generally.

In addition, Figure 4 and Figure 5 can also visually demonstrate the correctness and composition of the YOLO generated bounding boxes in the form of images.

3.3. Model convergence analysis

We first verify the convergence of the model, whose loss can be seen shown in Figure 1. For the training procedure, the box loss, obj loss and cls loss decrease with the increase of epochs and tend to be stable finally. We also can observe that the change of loss of verification broken lines are similar to those of training broken lines. The trend of the broken lines shows that the model has basically converged.

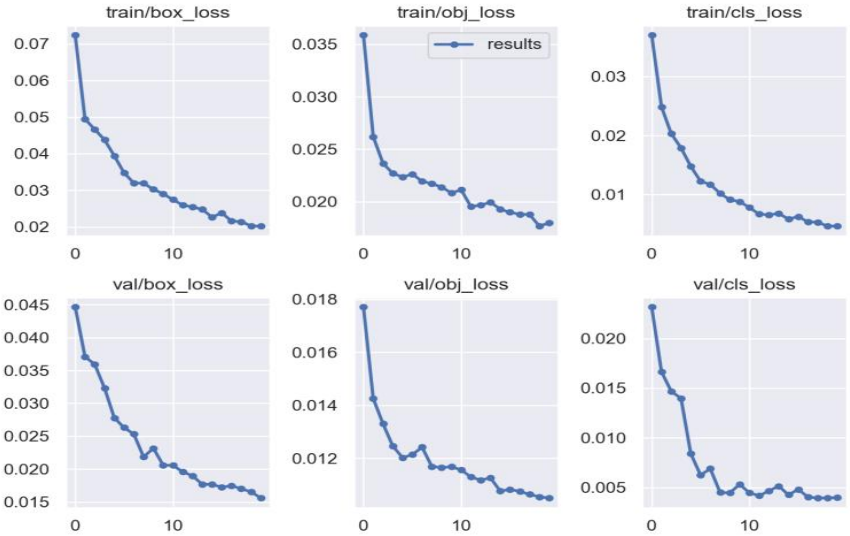


Figure 1. The change of loss with the increase of training procedure.

3.4. Quantitative analysis

We further analyzed the running results of the model to prove the practicality of it, which are reported in Figure 2. We can achieve a precision and recall more than 90%, and a mAP of 96% and mAP of 79% when the threshold are 0.5 and 0.5：0.95, respectively. To analyze the detection performance of various categories, we also give the confusion matrix of our method in Figure 3. The results show that we can achieve a precision of 90% for buffalo, precision of 98% for elephant and rhino, a precision of 94% for zebra. All the results show the effectiveness of our wlid animals detection model based on YOLOv5.

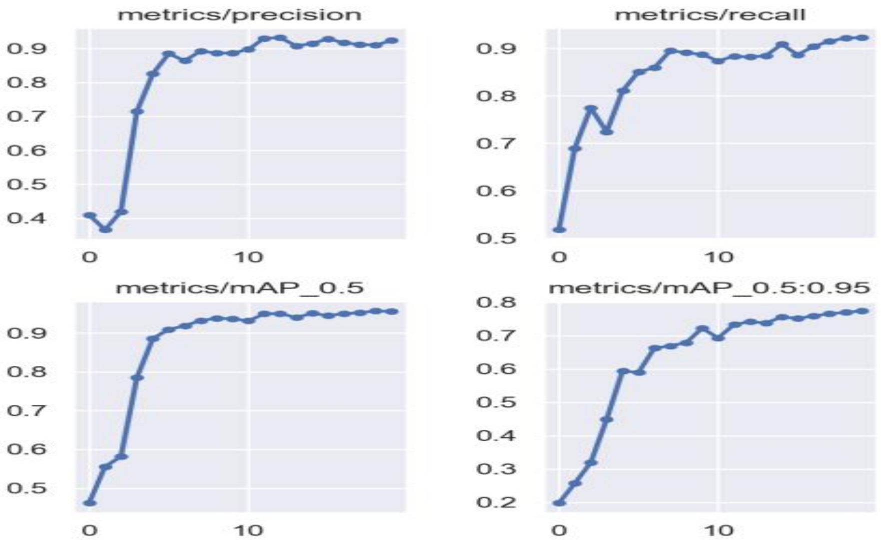


Figure 2. mAP of our module.

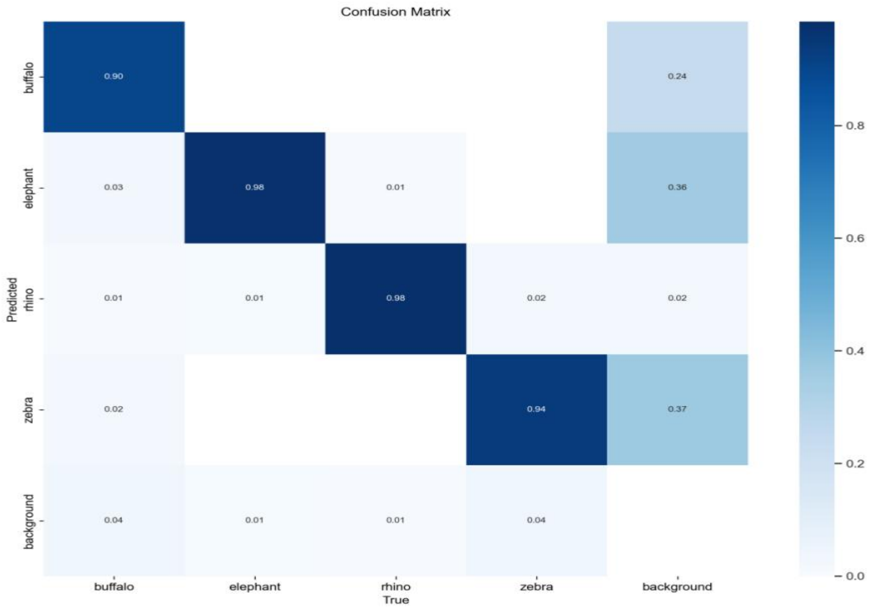


Figure 3. Confusion matrix of proposed method.

3.5. Visual analysis

We also visualize the detection results of different kinds of animals in various scenes, which can be seen in Figure 4. For different kinds of animals, YOLOv5 generates very accurate bounding boxes, which can basically fit the animal body boundaries. There are also good results for some quite complex images, such as multi-target images, and there is no obvious poor detection effect. In addition, YOLOv5 has an average prediction time of less than 200ms, and has the ability to detect real-time targets, if necessary, MobileNetv3 can also be introduced to further speed up [10]. The current accuracy and time consumption have basic application value.

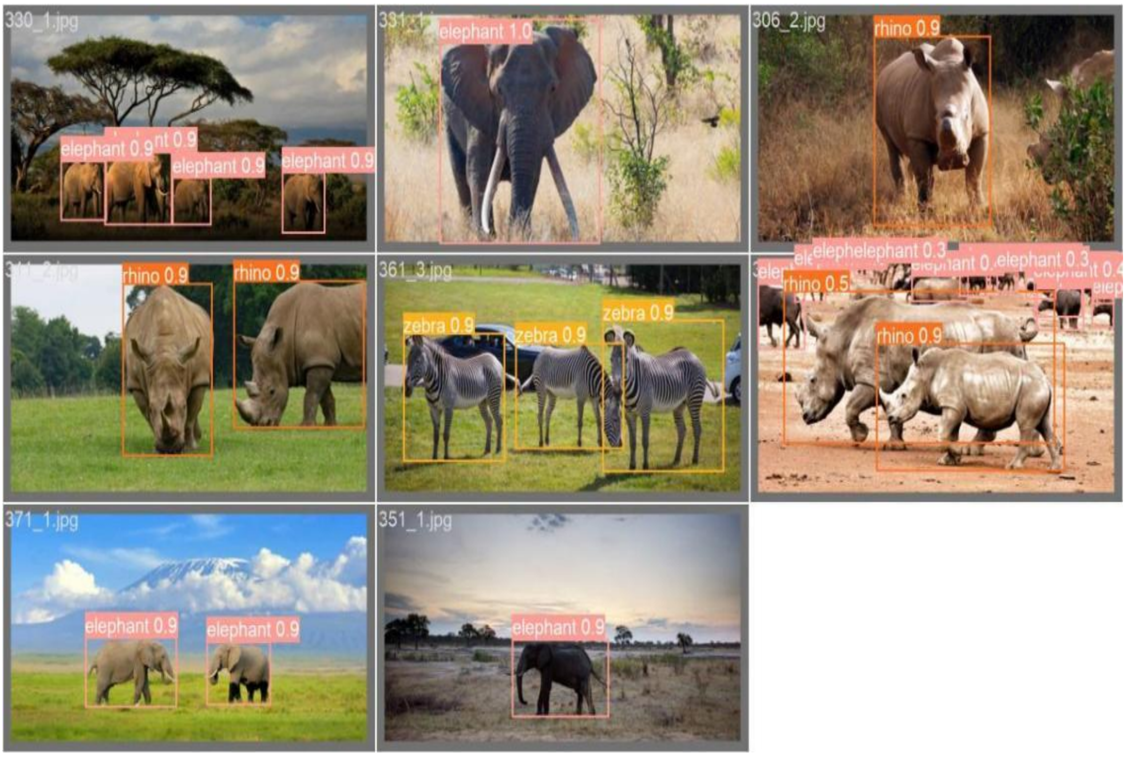
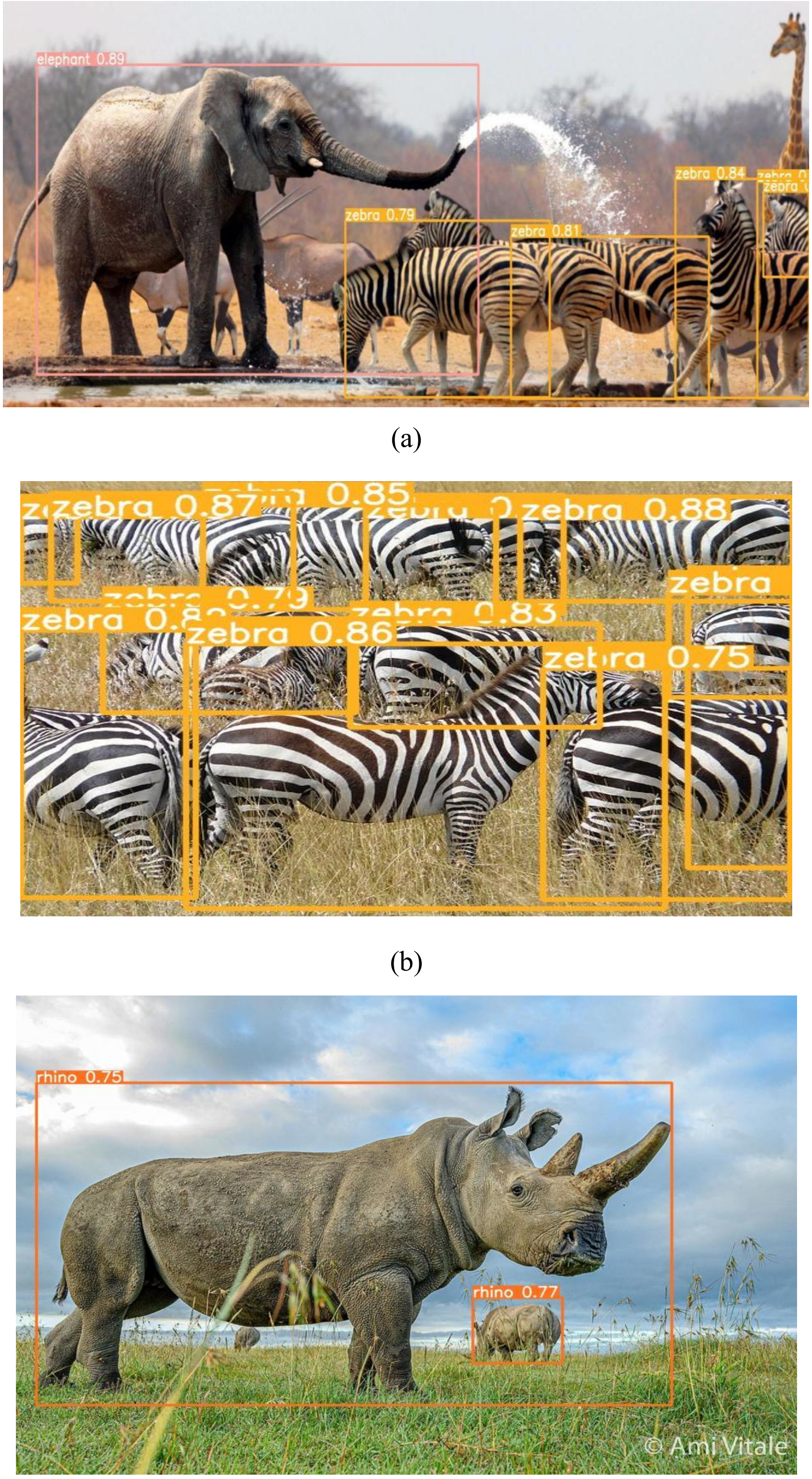


Figure 4. Detection results for different kinds of wild animals.

In order to further explore the practicability of YOLOv5 , three common cases are specially selected here for verification in Figure 5: multiple targets in a picture, part of the animal's body is blocked, and the detection target is the animal with a long distance from camera or small size. In the field, the frequency of these three situations is very high, and they will exactly interfere with the result of target detection to some extent, and then affect the practicability of model.

Obviously, YOLOv5 exhibits relatively good results in the face of dense targets. It can basically detect all animals under relatively dense conditions, and only a few animals are lost under very dense conditions. Besides, its accuracy is excellent when distinguishing animals whose bodies are covered by other objects. However, the detection effect of small targets needs to be improved. Some of the small animals in the picture have not been detected, and there are also small target detection errors.



(a)

(b)

(c)

Figure 5. Detection results for animals in different scenes.

4. Discussion

Through satisfactory detection performance can be obtained by our model, there are still some problems to be improved. The first issue is the data set not perfect, and some situations that may occur during field shooting do not have corresponding pictures to reproduce, such as shooting in the dark, or shooting from the perspective of unmanned aerial vehicle. The lack of these data sets may have a certain impact on the results of the model in similar cases, affecting the actual application effect. To overcome this problem, the data set should contains more specific types of pictures, especially with the help and guidance of people with field investigation experience, so that the data set can be as realistic as possible to explore various situations that may occur.

For small target detection, due to the actual situation, there is often a need for small target detection, especially for small animals. On the one hand, we need to increase the sample size of small targets, and on the other hand, we need to make improvements to the algorithm. The prediction can be made by a shallow neural network, which can reduce the range of pixels that affect the prediction [5], and can appropriately increase the weight ratio of the shallow network to the prediction results, so that the neural network can pay more attention to small targets [11]. According to the application situation, it can also sacrifice the discrimination speed to a certain extent, and provide special neural network for small targets, such as adding heuristic rules to simulate human strategy when dealing with small targets.

In addition, some very rare species of animals may lack observation records, resulting in too few samples used in training, or the sample quality is not high. The results of such training will be affected to a certain extent [12], but at the same time, rare animals always need to be found.

In this case, YOLOv5 can train both data sets of rare animals and other animals with similar appearance to rare animals at the same time. Because YOLOv5 supports multi-label detection, it can classify rare animals into other animals first to increase the detection accuracy. After the number of samples of this rare animal species increases, the detection can be independent.

5. Conclusion

In the face of the challenging needs and tasks of wildlife detection, this article provides a feasible method through Yolov5 algorithm. Specifically, we first revisit the theory of YOLOv5, including its network structure, loss function, training settings. We conduct several experiments to analyze whether this method is feasible, especially for different wild scenes, such as multiple targets, small targets or occluded targets. We have used a large number of experiments to verify the feasibility of this method and achieved a good result.

References

[1] Shivang A, Jean O D T and Frédéric D 2018. J. Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks. arXiv e-prints arXiv1809.03193.

[2] Ross G 2015. J. Fast R-CNN. 2015 IEEE International Conference on Computer Vision. 1440-1448.

[3] Carranza-García M, Torres-Mateo J, Lara-Benítez P and García-Gutiérrez J 2021. J. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. Remote Sens. 13(1): 89.

[4] Kanlayanee K, Siranee N and Joshua M P 2021. J. Open source disease analysis system of cactus by artificial intelligence and image processing. The 12th International Conference on Advances in Information Technology.

[5] Delong Q, Weijun T, Qi Y and Jingfeng L 2021. B. YOLO5Face: Why Reinventing a Face Detector. ECCV Workshops.

[6] Joseph R, Santosh D, Ross G and Ali F 2015. J. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE International Conference on Computer Vision and Pattern Recognition. 779-788.

[7] Joseph Rand Ali F 2018. J. YOLOv3: An Incremental Improvement. arXiv e-prints arXiv1804.02767

[8] Alexey B, Chien-Yao W and Hong-Yuan M L 2020. J. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv e-prints arXiv2004.10934

[9] Hu J, Zhi X, Shi T, Zhang W, Cui Y and Zhao S. 2021, J. PAG-YOLO: A Portable Attention-Guided YOLO Network for Small Ship Detection. Remote Sensing. 13(16):3059

[10] Hou Y, Yang Q, Li L and Shi G 2023. J. Detection and Recognition Algorithm of Arbitrary-Oriented Oil Replenishment Target in Remote Sensing Image. Sensors (Basel). 23(2):767.

[11] Munhyeong K, Jongmin J, and Sungho K 2021. J. ECAP-YOLO: Efficient Channel Attention Pyramid Yolo for Small Object Detection In Aerial Image. Remote Sensing, 13(23):4851.

[12] Madodomzi M, Philemon T, Tsungai Z and Abel R 2021. J. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. Remote Sens. 13(1):89.