Proceedings of 2023 IEEE
International Conference on Mechatronics and Automation
August 6 - 9, Harbin, Heilongjiang, China

YOLOv7-based land and underwater target detection and recognition

Jieren Li[1,3], Liwei Shi[1,3], Shuxiang Guo[1,2,3,4],*

1. *School of Medical Technology, Beijing Institute of Technology, Beijing, 100081, China*
2. *School of Life Science, Beijing Institute of Technology, Beijing, 100081, China*
3. *Key Laboratory of Convergence Medical Engineering System and Healthcare Technology(Beijing Institute of Technology), Ministry of Industry and Information Technology, Beijing, 100081, China*
4. *The Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen,Guangdong 518055,China*

**Corresponding author: shiliwei@bit.edu.cn*

***Abstract* - There is a growing demand for marine resource development. Bionic amphibious robots can replace humans to conduct land and underwater exploration, which has important research significance. Deep learning has developed rapidly in recent years, and many kinds of target detection algorithms have emerged. We select one two-stage target detection algorithm Faster R-CNN, three single-stage target detection algorithms SSD, Centernet, and YOLOv7. We use each of these four algorithms to train the VOC2007 dataset in a deep learning environment. After the training is completed, these four models are evaluated and predicted separately. We find an algorithm that is most suitable for the amphibious robot application—YOLOv7. Finally, we use the YOLOv7 model to detect the underwater dataset, and the results prove that the model is promising for detecting small underwater targets.**

Index Terms – Amphibious robots, YOLOv7, SSD, Centernet, Faster R-CNN.

I. INTRODUCTION

At present, there has been an increasing demand for marine resource exploration, search and rescue at sea. As a new type of key industrial equipment, amphibious robots have obvious advantages in terms of motion mobility performance, manufacturing and maintenance costs, etc. Amphibious robots can adapt to land, underwater and land-water transition environments[1-2]. Amphibious robots have a very wide range of application scenarios. Robot vision is a crucial aspect of robot, which determines whether the robot can judge the surrounding environment well and recognize objects in the environment[3]. In addition, target tracking is also a crucial aspect of robotics. Traditional target detection algorithms rely on delicate manual feature design and extraction, and the accuracy is somewhat limited, which is poor for the detection of small amphibious robots and the recognition of complex environments[4-5]. There are many researchers working on bionic amphibious robots and underwater robots[6-9].

(a) AmphiBot[10] (b) Whegs[11]

Fig. 1. Amphibious robot

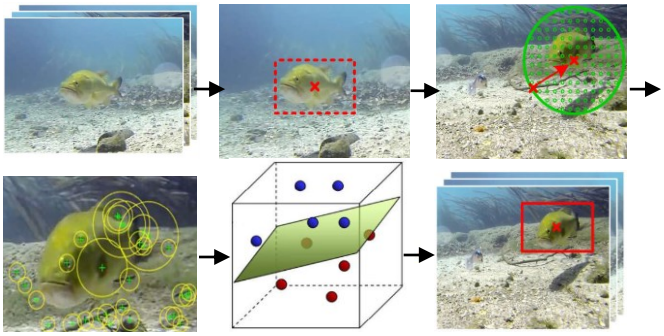

Fig. 2. Basic framework of image target tracking algorithm[12]. Video or image sequences, Target initialization, Motion prediction and candidate sample collection, Target feature extraction, Statistical modeling and updating of target features, Target tracking results.

With the advent of the era of deep learning, target detection algorithms based on deep learning are more accurate and faster than traditional algorithms, and gradually come into the public eye[13]. With the gradual development of hardware systems such as equipment, the technology has become more and more mature, and the problems faced by deep learning, such as unusually large amount of data and long training time, have been effectively solved, and the target detection and recognition technology based on deep learning has developed rapidly. This paper focuses on the current mainstream deep learning-based target detection algorithms, and aims to screen out a target detection algorithm suitable for amphibious robots for robot target identification and tracking.

The rest of the paper is structured as follows. Section II introduces single- and two-stage target detection algorithms based on deep learning. Section III describes the algorithm environment construction and dataset construction. Section IV performs comparison and selection of target detection algorithms. Section V describes YOLOv7 detection of underwater datasets. Section VI describes the conclusion and future work.

II. OBJECT DETECTION ALGORITHM BASED ON DEEP LEARNING

Deep learning target detection algorithms are classified into single-stage target detection and two-stage target detection algorithms.

A. Two-stage target detection algorithm

The two-stage target detection algorithm is a "coarse-to-fine" process, in which the algorithm generates target candidate frames in the first stage and performs category classification and border regression on the candidate frames in the second stage. R-CNN (region-based convolutional neural network) [14] is a typical two-stage algorithm, in addition, there are also improved versions such as Fast R-CNN , Faster R-CNN [15] networks, etc. The Faster R-CNN network uses a convolutional layer to extract image features; the RPN network extracts candidate regions based on the input image features; uses an ROI (Region of Interest) pooling layer to transform to a fixed-length output; and finally, classification is performed using a classification regression layer. All tasks of this approach are under a single deep learning framework, with substantially faster computation and higher accuracy.

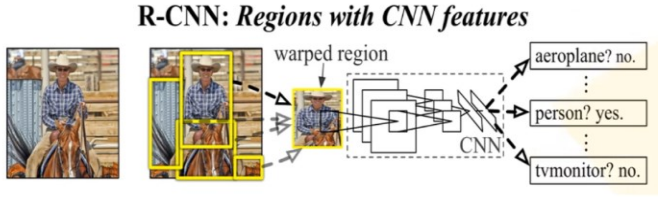

Fig. 3. Network structure of Faster R-CNN[15]

B. One-stage target detection algorithm

The single-stage target detection algorithm is different from the two-stage target detection algorithm, which does not require the stage region recommendation and directly generates the category probability and location coordinate values of the object. After single-stage detection, the final detection results are obtained directly available and therefore have a faster detection speed.

The YOLO (You Only Look Once) [16] algorithm was proposed by Redmon et al in 2015. The input image requires only one network computation to directly obtain the target bounding box and category probability in the image. The YOLO family of algorithms is an advanced one-stage target detection algorithm framework that has been successfully applied to machine vision tasks such as traffic , medical, and industrial inspection. The YOLO is under continuous optimization and has now evolved to YOLO v7.

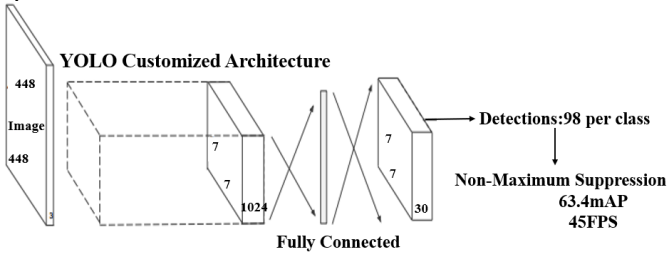

Fig. 4. Network structure of YOLO

The SSD (Single Shot MultiBox Detector) [17] algorithm was improved by Liu et al. on YOLOv1. The detection accuracy and detection speed are greatly improved with respect to YOLOv1. The core of SSD is the use of a small convolutional filter applied to the feature map to predict the class scores and box offsets for a fixed set of default bounding boxes. These design features result in simple end-to-end training and high accuracy. Speed and accuracy are improved even on low-resolution input images.

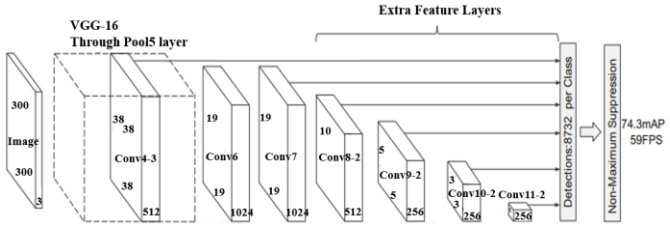

Fig. 5. Network structure of SSD

Centernet [18] is an anchor-free target detection network that models the centroid of an object bounding box, uses keypoint estimation to find the centroid, and regresses to all other object properties, such as size, 3D position. The network obtains center heatmap and corner heatmaps by center pooling and cascade corner pooling respectively, which are used to predict the location of keypoints. After getting the positions and classes of corner points, the positions of the corner points are mapped to the corresponding positions of the input image by offsets, and then the embedings are used to determine which two corner points belong to the same object in order to form a detection frame.

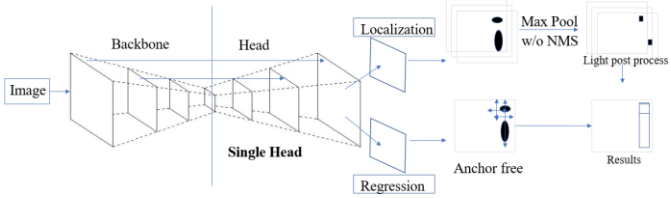

Fig.6. Network structure of Centernet

III. ALGORITHM ENVIRONMENT CONSTRUCTION AND DATA SET CONSTRUCTION

In this paper, we use Windows 11 operating system, the programming environment is virtual environment python=3.9, the deep learning framework is pytorch2.0.0, the NVIDIA GPU acceleration platform is cu117, and the GPU hardware is NVIDIA RTX3060 (12G).

A. Introduction of important dependency libraries

The important dependencies used in the implementation of the algorithm are shown in Table I.

TABLE I
DEPENDENCY LIBRARY LIST

Dependency Libraries	Version
Numpy	1.24.1
Matplotlib	3.7.1
Opencv	4.7.0.72
Scipy	1.10.1
Torch	2.0.0+cu117
Tensorboard	2.12.0
Torchvision	0.15.1+cu117

Numpy is a library of extensions for Python, mainly for efficient operations on arrays and matrices[19]. Numpy also serves as the basis for many other libraries and can be used

with a large number of extensions. Today's most popular Tensorflow and Pytorch frameworks also use Numpy to process data, so it is indispensable in today's scientific computing, especially in the field of machine learning.

Matplotlib is a python library for drawing graphs and charts[20]. It can be used with the Numpy library to get icons similar to those in MatLab. In this project, we use Matplotlib to plot loss function drop, AP, F1, MAP, etc.

Opencv is an open source computer vision library[21]. Through Opencv we can easily read images, transform, save and other operations. It contains a variety of image processing algorithms interface, can be used with the Numpy library. thus making the processing of images easier. In this project, we use Opencv to read and pre-process images.

Scipy is a library of algorithms for a wide range of mathematical calculations and engineering applications[22]. It has a variety of built-in functions for solving ordinary differential equations, performing interpolation and integration operations. It can be used together with Numpy matrices to improve computational efficiency.

B. Dataset construction

Image information is rich in content, and the target is clear, which can improve the resolution, so it is often used for target recognition. We start by collecting images or videos of the target and organizing them into a dataset. General images or videos can be captured in water by underwater vehicles or fixed camera equipment, or they can be collected via the Internet. We first use the public dataset VOC2007[23] to predict.

Annotations holds the annotation information of the images. Main is the list of datasets for target detection, and JPEGImages holds the original images.

IV. Object Detection Algorithm Screening and Implementation

In this paper, four deep learning-based target detection networks are built using the pytorch deep learning framework. They are respectively: two-stage target detection network faster-rcnn, single-stage target detection network SSD, CenterNet, and YOLOv7. These four target detection algorithms are used to train the VOC2007 dataset separately, and the training and validation sets are divided in the ratio of 9:1.

A. Evaluation indicators

Evaluation *indicators* for target detection usually use Precision, Recall, F1 score (F1), Average Precision (AP), Mean Average Precision (mAP), etc.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \frac{\sum_{j=1}^{N} \frac{1}{i} \sum_{i=1}^{K} Precision(i)}{N} \tag{3}$$

$$F = \frac{2Precision * Recall}{Precision + Recall} \tag{4}$$

TP refers to true positive, TN refers to true negative, FP refers to false positive, FN refers to false negative.

B. Faster R-CNN Training and Evaluation

The Faster R-CNN model uses the Adam optimizer and sets the maximum learning rate to 1e-4. The learning rate is decreased by "cos", and the weights are kept in the "logs" file once every 5 training epochs. The total number of training epochs is 100. In order to speed up the training and to prevent the network weights from being destroyed at the beginning of the training, 50 rounds of training are frozen first and the next 50 epochs are unfrozen. During the training process, the mAP of the validation set is evaluated every 5 epochs, and the appropriate batch_size is selected according to the complexity of the network structure, the memory consumption and the training phase.

When training the Faster R-CNN model, the batch_size for the freeze training phase is set to 8 and the batch_size for the unfreeze training phase is set to 4.

C. SSD Training and Evaluation

SSD adopts VGG16 as the base network. The core design concepts of SSD are: 1. Using multi-scale feature maps for prediction to improve the accuracy of recognition. 2. Drawing on the anchor concept in Faster R-CNN, each unit sets Default boxes with different scales or aspect ratios, and the predicted bounding boxes are based on these Default boxes. This reduces the training difficulty to a certain extent. 3. Convolution is directly used to extract detection results for different feature maps.

The SSD model uses the SGD optimizer and sets the maximum learning rate to 2e-3. The weights decay to 5e-4 to prevent overfitting. The learning rate decreases by "cos" and the weights are kept in the "logs" file every 10 training rounds. The total number of training rounds is 200. In order to speed up the training and prevent the network weights from being destroyed at the beginning of the training, 50 rounds of training are frozen first, and then 150 rounds are unfrozen. During the training process, the mAP of the validation set is evaluated every 10 rounds, and the appropriate batch_size is selected according to the complexity of the network structure, the amount of memory used, and the training phase.

The batch_size for the freeze training phase is set to 128 and the batch_size for the unfreeze training phase is set to 64.

D. Centernet Training and Evaluation

The Centernet model uses the Adam optimizer and sets the maximum learning rate to 5e-4. The learning rate is decreased by "cos". The weights are kept in the "logs" file every 5 training rounds. The total number of training rounds is 100. In order to speed up the training and prevent the network weights from being destroyed, at the beginning of the training,

50 rounds of training are frozen and the next 50 rounds are unfrozen. During the training process, the mAP of the validation set is evaluated every 5 rounds, and the appropriate batch_size is selected according to the complexity of the network structure, the memory consumption and the training phase.

The batch_size for the freeze training phase is set to 64 and the batch_size for the unfreeze training phase is set to 32.

E. YOLOv7 Training and Evaluation

YOLOv7 is currently the most advanced algorithm in the YOLO series, surpassing the previous YOLO series in terms of accuracy and speed.

The YOLOv7 model uses the SGD optimizer, sets the maximum learning rate to 1e-2. To prevent overfitting, sets the weights to decay to 5e-4. The learning rate decreases by "cos" and keeps the weights in "logs" every 10 training rounds. The total number of training rounds is 300. In order to speed up the training and prevent the network weights from being destroyed at the beginning of the training, 50 rounds of training are frozen first, and then 250 rounds are unfrozen. During the training process, the mAP of the validation set is evaluated every 10 rounds, and the appropriate batch_size is selected according to the complexity of the network structure, the amount of memory used, and the stage of training.

The batch_size for the freeze training phase is set to 16 and the batch_size for the unfreeze training phase is set to 8.

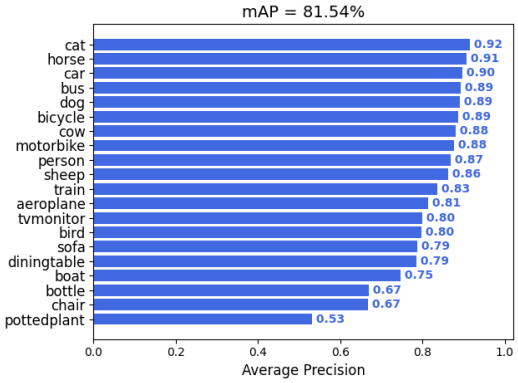

Fig.7. mAP of Faster R-CNN

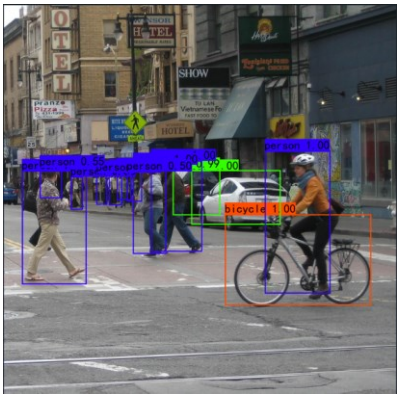

Fig.8. Prediction results of Faster R-CNN

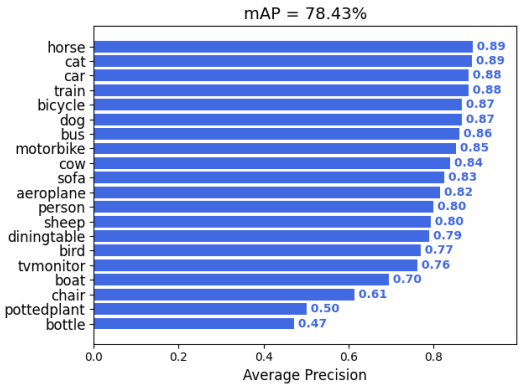

Fig.9. mAP of SSD

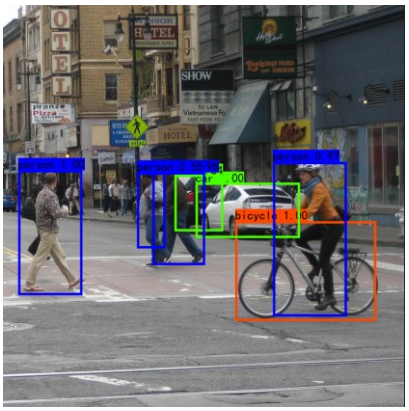

Fig.10. Prediction results of SSD

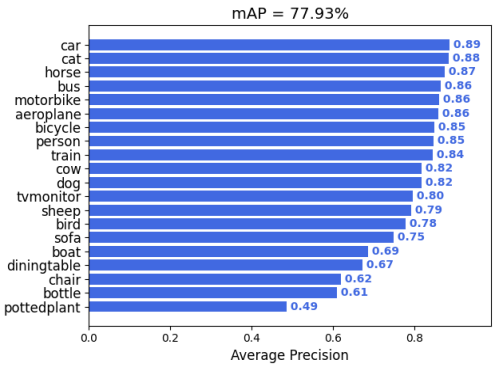

Fig.11. mAP of Centernet

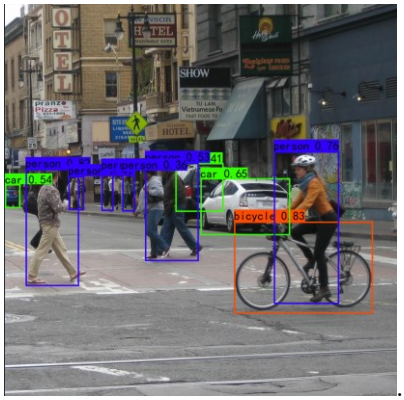

Fig.12. Prediction results of Centernet

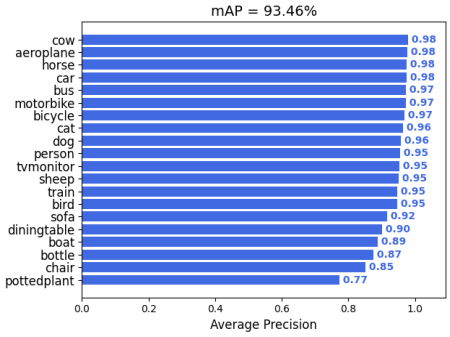

Fig.13. mAP of YOLOv7

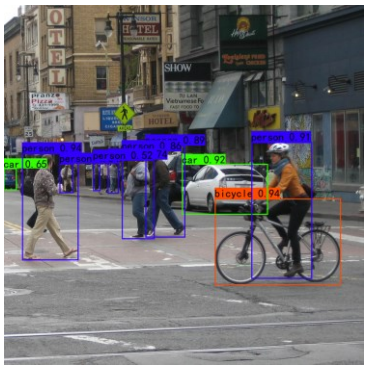

Fig.14. Prediction results of YOLOv7

F. *Comparison of the results of each model*

TABLE II
AP COMPARISON BY MODEL

AP(%)	aeroplane	bicycle	bird
YOLOv7	97.67	96.85	94.62
SSD	81.52	86.83	77.11
FasterRCNN	81.42	88.59	79.66
Centernet	85.84	84.91	77.87

TABLE III
F1 COMPARISON BY MODEL(SCORE_THREHOLD=0.5)

F1	aeroplane	bicycle	bird
YOLOv7	0.95	0.95	0.90
SSD	0.80	0.83	0.77
FasterRCNN	0.64	0.78	0.71
Centernet	0.76	0.73	0.63

TABLE IV
PRECISION COMPARISON BY MODEL

PRECISION(%)	aeroplane	bicycle	bird
YOLOv7	98.15	97.21	93.19
SSD	85.32	86.93	87.19
FasterRCNN	51.27	69.11	62.56
Centernet	98.88	98.01	95.96

TABLE V
RECALL COMPARISON BY MODEL

RECALL(%)	aeroplane	bicycle	bird
YOLOv7	92.98	93.18	86.49
SSD	75.44	78.93	68.19
FasterRCNN	84.91	89.61	80.83
Centernet	61.75	58.46	46.84

The mAP value of YOLOv7 is 93.46, which is the highest.

The indicators results of the four models for the detection of the VOC2007 dataset are shown in Table II-V. We have chosen three categories to compare: aeroplane, bicycle, bird.

The above tables show that the precision of YOLOv7 algorithm is slightly worse than Centernet, but the other parameters are better than SSD, Centernet, Faster R-CNN.

G. *Algorithm Comparison Analysis*

Since Faster R-CNN is a two-stage detection algorithm, it is more accurate, but the detection speed is very slow and not suitable for real-time detection. Therefore, we need to choose a single-stage detection algorithm. SSD fuses the features of different convolutional layers, thus achieving multi-scale target detection, so it is better for small target detection. CenterNet is an anchor-free target detection network, which is more advantageous in terms of speed and accuracy. In YOLOv7 a training method of auxiliary head is proposed, with the main purpose of improving the accuracy by increasing the training cost without affecting the inference time, since the auxiliary head will only appear during the training process. E-ELAN is used on the backbone network instead of the original CSPDarknet53, and the SPP module is redesigned. In the head part, YOLOv7 adds RepConv, which has 3 branches for 1×1, 3×3 convolution and BN during training, and the convolution and BN of the 3 branches can be equivalently fused to form a 3×3 convolution of VGG structure during model deployment, thus speeding up the model inference.

When YOLO makes a prediction, it reasons about the image in a comprehensive way. During training and testing, the entire image is seen, and YOLO has half the number of background false positives than R-CNN.

After analyzing the structure and principles of the four algorithms, we found that YOLOv7 can meet our usage requirements in terms of accuracy and precision. YOLOv7 is also the most suitable algorithm for bionic amphibian robot application scenarios, and performs better in natural environments where amphibian robots work, such as: mud flats, underwater, etc.

V. YOLOv7 DETECTION OF UNDERWATER DATASETS

We selected an algorithmic network suitable for target detection: YOLOv7, and we next employ this model to detect the underwater dataset and observe the detection effect.

The underwater target dataset was adopted from a real collection of underwater target images provided by Dalian University of Technology in the underwater target detection algorithm competition (optical image competition). The detection objects in the dataset are mainly economic seafood in real waters. The shooting time was 8:00-11:00 a.m. and 1:00-4:00 p.m. The shooting depth was between 0.5 and 8 meters due to tidal variations in the sea. Due to the variation of water depth and shooting time, the background of the sea water showed three shades of blue, green and blue-green.

There is no inter-frame continuity between these images. Four types of targets are detected in the dataset: holothurian, echinus, scallop, and starfish. We use this underwater dataset to train the model and make predictions. Figure 15-16 shows the prediction results.

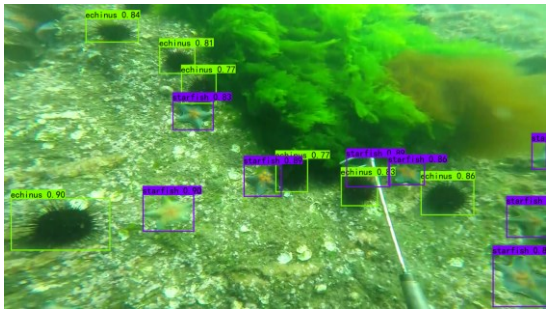

Fig.15. Underwater image prediction results

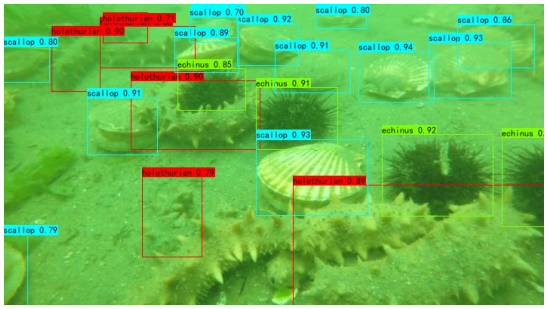

Fig.16. Underwater image prediction results

VI. CONCLUSION AND FUTURE WORK

The purpose of this paper is to find a suitable algorithm for target detection and recognition of a bionic amphibious robot. We have selected four currently dominant algorithms: Faster R-CNN, SSD, Centernet, YOLOv7. We use these algorithms to train public dataset VOC, and get the predict results. Then we use some indicators to evaluate the results of each model. Finally we find a most suitable algorithm for the amphibious robot—YOLOv7. In the next step, We shoot and build our own land and underwater datasets to train the yolov7 model. Then try to use it for the vision system of a bionic amphibious robot. After the detection work is completed, we will perform multi-objective tracking to supervise and predict the robot's motion trajectory. In addition, we will seek a suitable method to integrate the target detection and multi-target tracking system into the bionic amphibious robot to achieve the purpose of real-time detection and tracking of the target of the bionic amphibious robot.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (62273042, 61773064).

REFERENCES

[1] Z. Wu, J. Qi, S. Zhang, "Amphibious robots: a review," *Applied Mechanics & Materials*, vol. 2963, no. 494-495, pp. 1036-1041, February 2014.

[2] H. Xing, Y. Liu, S. Guo, L. Shi, X. Hou, W. Liu, Y. Zhao, "A Multi-Sensor Fusion Self-Localization System of a Miniature Underwater Robot in Structured and GPS-Denied Environments," *IEEE Sensors Journal*, vol. 21, no. 23, pp. 27136-27146, October 2021.

[3] G. Dudek, P. Giguere, C. Prahacs, S. Saunderson, J. Sattar, LA. Torres-Mendez, M. Jenkin, A. German, A. Hogue, A. Ripsman, "AQUA: An amphibious autonomous robot," *Computer*, vol. 40, no. 1, pp. 46-53, January 2007.

[4] L. Shi, P. Bao, S. Guo, Z. Chen and Z. Zhang, "Underwater Formation System Design and Implement for Small Spherical Robots," *IEEE Systems Journal*, vol. 17, no. 1, pp. 1259-1269, March 2023.

[5] L. Shi, Z. Zhang, Z. Li, S. Guo, S. Pan, P. Bao, L. Duan, "Design, Implementation and Control of an Amphibious Spherical Robot," *Journal of Bionic Engineering*, vol. 19, no. 6, pp. 1736-1757, July 2022.

[6] D. Low, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November 2004.

[7] H. Xing, S. Guo, L. Shi, X. Hou, Y. Liu, H. Liu, Y. Hu, D. Xia and Z. Li, "A novel small-scale turtle-inspired amphibious spherical robot," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1702-1707, 2019.

[8] L. Shi, S. Guo, K. Asaka, "A novel jellyfish-and butterfly-inspired underwater microrobot with pectoral fins," *International Journal of Robotics and Automation*, vol. 27, no. 3, pp. 276-286, 2016.

[9] L. Shi, S. Guo, K. Asaka, "A novel multifunctional underwater microrobot," *2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 873-878, 2010.

[10] A. Crespi, A. Badertscher, A. Guignard, "AmphiBot I: an amphibious snake-like robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163-175, 2016.

[11] A. Boxerbaum, "A whegs robot featuring a passively compliant, actively controlled body joint" *Cleveland, USA: Case Western Reserve University*, 2010.

[12] N. Wang, J. Shi, DY. Yeung, J Jia, "Understanding and diagnosing visual tracking systems," *Proceedings of 2015 IEEE International Conference on Computer Vision (CVPR 2015), New York, USA: IEEE*, pp. 3101-3109, 2015.

[13] S. Guo, S. Pan, X. Li, L. Shi, P. Zhang, P. Guo, Y. He, "A system on chip-based real-time tracking system for amphibious spherical robots," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, pp. 1-19, July 2017.

[14] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, June 2017.

[15] Y. Li, S. Guo, and C. Yue, "Preliminary concept and kinematics simulation of a novel spherical underwater robot," IEEE International Conference on Mechatronics & Automation, pp. 1907-1912, August 2015.

[16] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Computer Vision & Pattern Recognition*, 2015.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg, "SSD: Single Shot MultiBox Detector," *Computer Vision & Pattern Recognition*, 2015.

[18] T. Zhang, G. Wang, Y. Zhuang, H. Chen, L. Chen, "Feature Enhanced Centernet for Object Detection in Remote Sensing Images," *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020.

[19] C. Oswald, "Python Video Tutorials: NumPy and Pandas," *Mercury Learning and Information*, 2021.

[20] L. Roubeyrie, S. Celles, "Windrose: A Python Matplotlib, Numpy library to manage wind and pollution data, draw windrose," *Journal of Open Source Software*, vol. 3, no. 29, pp. 268-268, 2019.

[21] N. Abdul, S. Abhishek, T. Ashutosh, P. Ayushi, "Face Mask Detection Using OpenCV," *Advances in Science and Technology*, vol. 6630, pp. 53-59, 2023.

[22] J. Charles, "Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib," *Siam Review*, vol. 62, no. 2, pp. 515-517, 2020.

[23] Anonymous, "CX Program Success: Understanding the Real Role of VOC Software," *Customer Relationship Management*, vol. 26, no. 9, pp. WP2-WP4, 2022.