ROV TARGET GRASPING STRATEGY BASED ON VISUAL PERCEPTION

Jiawen Li,* Xiang Cao,** and Xueyou Huang*

Abstract

With the increasing prominence of unmanned systems in aquaculture and fisheries, this study proposes a visual perception-based remote operated vehicle (ROV) target grasping strategy to address complex underwater environments and high operational risks. The strategy consists of two parts: target detection and target grasping. Target detection is to use the visual sensor carried by ROV to perceive the underwater environment, obtain environmental images, and use deep neural network algorithms to detect targets in the images. The detected target is the premise of realising the target grasping, and the target grasping uses a fuzzy PID algorithm to control the robot arm carried by the ROV to grasp the detected target. Simulation and experiments show that this method can realise target detection and grasping under different water quality and has higher detection accuracy and speed. In practical engineering applications, this method meets the requirements of intelligent aquatic fishing in complex underwater environments.

Key Words

ROV, YOLO, target detection, target grasping

1. Introduction

Currently, traditional fishing methods are used to harvest aquatic products like sea cucumbers and sea urchins. However, these methods are not always efficient in completing fishing tasks. To solve this issue, this paper presents a visual perception-based strategy for ROV target grasping. This method is designed to adapt to the complex and dynamic underwater environment and is both safe and reliable. Additionally, it is highly portable. The strategy is comprised of two main modules: target detection and target grasping. The two are interrelated and cannot be separated. The target detection algorithm detects underwater target objects and determines their location, category, and other information. The target grasping task formulates grasping strategies based on this information. Embedding object detection and grasping algorithms into the ROV control system enables it to make decisions and adjustments in a real-time environment. In conclusion, target detection and grasping are complementary tasks, and their close connexion and mutual influence make real-time grasping tasks more effective in practical applications. The results of the two tasks are discussed separately, which can improve ROV's perception and operation capabilities in complex environments [1]–[3]. Each module will be introduced in detail below.

Target detection is a fundamental problem in computer vision, and its purpose is to identify all of the targets of interest in an image and establish their class and location. Since all kinds of different objects have a different appearance, pose, and different degree of occlusion, and the imaging is disturbed by lighting and other factors, target detection has been a very challenging problem [4], [5]. The classic target detection algorithm and the depth learning algorithm are the two main phases in target detection. Target detection is traditionally divided into three parts. The first step is to locate the target using region selection and then to navigate the entire image using the sliding window. In the second step, the selected region is extracted. The third step is to use the model which has been trained in advance to carry on the region recognition. The traditional target detection has shortcomings in region selection. Because the target could appear at any point in the image and its size and aspect ratio are unknown, the sliding window method is used to traverse the entire image at first, and several scales with different orientations must be established. The speed and effectiveness of subsequent feature extraction and classification are significantly impacted by this thorough method's high temporal complexity and huge number of duplicated windows, which is especially problematic for real-time target identification and ROV grasping. Deep learning-based target recognition systems have quickly advanced due to the development of technology like graphics processing units [6]. "Two-Stage" and "One-Stage" are the two main divisions of deep learning-based target identification techniques. In 2014, Girshic *et al.* [7] proposed the region-convolutional neural network (R-CNN) model, which is the first attempt of a convolutional neural network in target detection, and it is based on a selective search of region selection box

* Anhui University, Institutes of Physical Science and Information Technology, Hefei 230039 China; e-mail: {lijiawen790, huangxueyou108}@163.com

** Anhui University, School of Artificial Intelligence, Hefei 230039 China; e-mail: xiangcao@cahu.edu.cn
Corresponding author: Cao Xiang

Recommended by Jiankun Wang
(DOI: 10.2316/J.2024.206-1062)

and convolutional neural network combined to improve the capability of feature representation, which makes the field of target detection enter a new stage. Redmon *et al.* [8] proposed the YOLO detection algorithm approach, where the idea of YOLO detection differs from that of the R-CNN family in that it solves the target detection as a regression task. To complete an end-to-end real-time target detection task, the YOLO approach employs a single convolutional neural network that utilises all of the visual input. This network predicts the target's bounding boxes while also determining the target's class. Full graph information can be used throughout the training and prediction phases of YOLO, which leverages it for prediction. Due to the Fast R-inability CNN's to examine the entire image during the identification process, it wrongly identifies background patches as targets [9]. The YOLO background prediction error rate is half as low as Fast R- CNN's. YOLO is a technique for real-time target identification based on a special neural network model that has characteristics that allow rapid detection with high accuracy and some stability. Later, with improvements, the team put forth YOLOv2 and YOLOv3 [10], which offer a lot of advantages in terms of detection accuracy. The single shot multibox detector (SSD) detection algorithm, put out by Liu *et al.* [11], has significant improvements in detection speed and accuracy, but has limitations in the detection of small objects. To increase the accuracy of target detection in complicated underwater environments, Wang *et al.* [12] developed an enhanced SSD-based target detection algorithm with ResNet instead of VGG network structure, however, it has drawbacks in real-time detection. Han *et al.* [13] proposed a new underwater target detection technique that can facilitate ROV underwater target classification, but there are defects in detection accuracy and it is difficult to perform real-time target detection and grasping. In summary, it can be seen that the target detection algorithms studied by previous researchers combined with ROV real-time target detection and grasping have defects, such as low real-time, low accuracy, and low detection speed.

After the target is identified, proceed with target grasping. Autonomous grasping means that without human intervention, the vision robotic arm system acquires the position of the target object through the camera and drives the robotic arm to complete the grasping task for the target object [14]–[16]. The robotics field faces many challenges in gripping technology. The wide range of applications of robotic arms has also contributed to the diversity of robotic arms. There are also many challenges that need to be solved urgently. For example, the ability to grasp accurately complex environments, the occlusion of complex backgrounds, and the collision of robotic arms [17]. The remaining networks were suggested by Trottier *et al.* [18] for object localisation utilising vision sensors to gather data for controlling a robotic arm for gripping activities. They employed a global average pooling layer before a fully connected layer to enhance the convolution step by eradicating the spatial correlation of the backpropagation error signal. They do not use pre-training but perform direct data expansion to avoid the overfitting phenomenon, and this approach largely reduces the training time. Nguyen *et al.* [19] used a Kinect camera for target recognition and estimated object grasping strategies from local area point clouds. Walker [20] proposed to represent the robot arm position in joint coordinates or Cartesian coordinates. A collection of model-free deep reinforcement learning algorithms that can resolve grasping problems in a range of complicated scenarios were proposed by Quillen *et al.* [21] after applying deep reinforcement learning algorithms to robot grasping. The algorithm proposes a simulated benchmark for robot grasping, evaluates the benchmark task based on various Q-function estimation methods, uses a deep neural network model for robot grasping, and combines the evaluation to select a suitable method for grasping. This method is validated in different robots and can achieve the grasping of complex target objects. Jebelli *et al.* [22] proposed a learning algorithm that neither requires a 3D model of the object by visually grasping the object seen for the first time, by building a logistic regression model, given an image of the object, the algorithm will try to identify in each image the points corresponding to the better position of the grasped target object. After triangulating this collection of points, a 3D position is ultimately obtained, and an attempt is made to grasp that area. Synthetic pictures are employed as the training set, and supervised learning is used to train and detect the grabbing point from the image. Finally, practical validation is carried out, and a range of items are successfully grabbed. The above researchers have proposed relevant grasping strategies to perform target grasping, but there is no portability in the deployment into embedded devices, which is only suitable for specific grasping tasks and cannot perform grasping tasks with high quality in the face of target grasping in complex underwater environments.

In response to the above issues, this paper proposes a target-grasping method based on visual perception. First, the camera on the ROV is used to obtain images of the underwater environment, and the YOLOv5 algorithm is used to automatically detect the position and category information of the target object. Then, the detected target position and category information are transmitted to the fuzzy PID controller, which controls the movement of the ROV manipulator by calculating and outputting control commands [23]. After the ROV performs the grabbing action, it monitors the grabbing results and returns information, which will be passed to the target detection module to optimise the target detection algorithm. After the ROV performs the grabbing action, it monitors the grabbing results and returns information, which will be passed to the target detection module to optimise the target detection algorithm. With the help of the controller, the ROV robotic arm can accurately grasp the target object in the underwater environment and move it to the designated position. Throughout the study, target detection and grasping play an important role. Target detection and target grasping are different stages of the same thing. Target detection guides the capture strategy, and the grasp results are fed back to optimise the target detection results. Embedding target detection and grasping algorithms into the ROV control system enables it to make decisions and adjustments in a real-time environment. In actual operation, ROV can adjust the grasping strategy in real-time according to the target detection results and grasping feedback to cope with changing situations. Finally, through experiments and simulation verification, the control parameters are adjusted to achieve better target detection and grab effect. Simulation and experiments demonstrate that the proposed method can achieve target detection and grasping under different water qualities, with higher detection accuracy and speed. In practical engineering applications, this method meets the requirements for intelligent aquatic fishing in complex underwater environments. In this paper, the fishing problem of sea cucumbers and sea urchins in complex environment is studied, and the main contributions are as follows:

1) ROV target grasping strategy based on visual perception: This paper proposes a visual perception-based ROV target grasping strategy, which aims to simplify the underwater salvage process and significantly improve the grasping success rate. To cope with the harsh underwater environment and limited diving depth, advanced computer vision technology is utilized, especially the YOLOv3, YOLOv4, and YOLOv5 object detection algorithms. By combining these algorithms, our strategy can efficiently and accurately identify target objects, such as sea cucumbers and sea urchins in complex underwater scenes, thus meeting the needs of smart aquaculture.
2) Lightweight models and efficient real-time target detection: For practical application requirements, the lightweight model is selected as the target detection solution. These models have a smaller size, occupy less memory, and can achieve higher detection accuracy and speed when performing fishing tasks. By combining the YOLO series of algorithms, the technology proposed in this paper can be quickly deployed on ROVs and mobile devices to achieve real-time target detection and grasping tasks.
3) The optimisation effect of fuzzy PID control: In this paper, to further optimise the grasping strategy, the fuzzy PID controller is introduced as a control algorithm, which is optimised for the control arm of the ROV robot. By using fuzzy PID control, a more precise and stable control effect is achieved, and the success rate and efficiency of the grasping operation are improved. The application of this control strategy shows good control accuracy and robustness in actual engineering and further improves fishing efficiency.
4) Risk reduction and efficiency improvement: The strategy proposed in this paper reduces the risk of manual seafood harvesting while significantly improving operational efficiency. Using the fast YOLO series of algorithms, the ROV can efficiently identify targets in harsh underwater environments and limited diving depths, including sea cucumbers and sea urchins. A large number of simulations and actual test verifications have demonstrated the high stability and adaptability of the strategy, regardless of the underwater environment or the shape of the target, it can achieve accurate detection and grasping, confirming its practicability.

The remainder of the paper is organised as follows. Section II introduces the ROV model. Section III introduces the Algorithm. Section IV presents the simulation and experiment under various situations. Finally, Section V concludes the study and presents future work directions.

2. ROV Model

This part mainly establishes ROV from the Kinematic model and ROV system design framework.

2.1 Kinematic Model

This part mainly introduces how to develop the mathematical model of ROV, and analyse it from the kinematics and dynamics models. The ITTC and SNAME recommendation system are adopted to establish the inertial coordinate system, carrier coordinate system $(O - xyz)$, and carrier coordinate system $(L - \xi\eta\zeta)$. The inertial frame of reference is defined as the reference frame with a fixed point on the ground as the origin. In this paper, the origin L is established, with the positive direction of the ξ-axis indicating the movement direction of the ROV, the positive direction of the ζ-axis pointing towards the centre of the earth, and the positive direction of the η-axis perpendicular to the plane formed by the $\zeta\xi$. The positive direction of this coordinate system is determined by the right-hand rule. The origin O of the carrier coordinate system also called the motion coordinate system, is situated at the ROV's centre of gravity. The positive direction of the G-axis points forwards, the x-axis is perpendicular to the y-axis with the positive direction pointing to the starboard side of the ROV, and the z-axis is perpendicular to the plane formed by xy, with the positive direction also determined by the right-hand rule. The carrier coordinate system moves with the ROV. A schematic diagram of the ROV in the inertial and carrier coordinate systems is shown in Fig. 1. The establishment of these models is crucial in ensuring accurate target detection and grasping by the ROV in real-time.

The coordinates (x, y, z) of the origin of the carrier coordinate system in the inertial coordinate system represent the location data of the ROV during its underwater mobility. When the origins of the two coordinate systems coincide, the angle (ϕ, θ, ψ) between the inertial coordinate system and the carrier coordinate system represents the attitude information of the ROV, and the position information (x, y, z) and the attitude information (ϕ, θ, ψ) together constitute the pose information of the ROV, denoted by $\eta = [x, y, z, \phi, \theta, \psi]$. In this context, ϕ represents the roll angle of the ROV, θ represents the pitch angle of the ROV, and ψ represents the yaw angle of the ROV. Newton's second law does not apply in the carrier coordinate system, thus developing controllers in this system is not practical despite the spatial motion equation of the ROV being easier to define in this system. Therefore, a conversion between

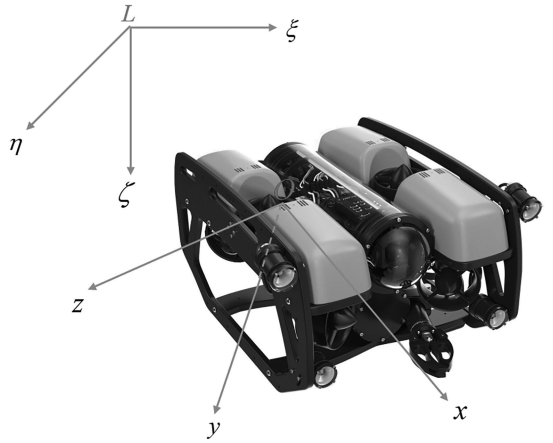

Figure 1. Inertial coordinate system and carrier coordinate system.

the carrier and inertial coordinate systems is necessary. After kinematic modelling, the transformation matrix $J(\eta)$ [shown in (1)] can be used to convert between the two coordinate systems.

$$J(\eta) = \begin{bmatrix} J_1(\eta) & 0_{3\times 3} \\ 0_{3\times 3} & J_2(\eta) \end{bmatrix} \tag{1}$$

In this paper, $J_1(\eta)$ represents the matrix used to transform linear velocity between the inertial and carrier coordinate systems, while $J_2(\eta)$ represents the matrix used to transform angular velocity between the two coordinate systems.

In this paper, the ROV is considered as a rigid body, and its motion in water is treated as the motion of a rigid body in a fluid. Using the Newton–Euler motion equations and the Lagrange modelling framework, the dynamic equations of motion of the ROV in water in the carrier coordinate system are given by (2).

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + G(\eta) = \tau w + \tau \tag{2}$$

M stands for the inertial matrix of the ROV with added mass, $C(\nu)$ for the matrix with added mass and centripetal and Coriolis forces, $D(\nu)$ for the hydrodynamic damping matrix, $G(\eta)$ for gravity and the restoring force vector, τw for the external disturbance vector, and τ for the torque vector brought about by the tension of the cable.

2.2 ROV Hardware System Architecture

The system hardware framework and communication interface, according to the distribution of robot functions and tasks, the system is designed as three control cores, the underlying STM32 driver board, Raspberry Pi, and shore computer. The rest also includes power system, camera, lighting, various sensors, data communication transmission system, *etc.* Connect the Raspberry Pi to the STM32 *via* USB to TTL, and the STM32 is connected to the ESC to control the rotation of the thruster motor. The PWM wave is calculated in the Raspberry Pi and transmitted to the STM32 through the serial port. The motion control part of the module runs on the Raspberry Pi. As shown in Fig. 2, the specific hardware system frame diagram.

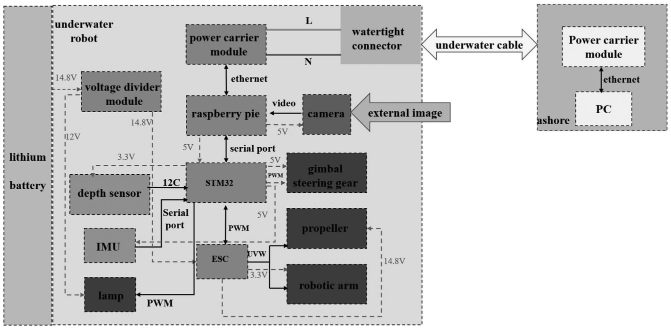

Figure 2. ROV hardware system architecture.

3. Algorithm

3.1 Target Detection Strategy

Target detection strategy is one of the focuses in this paper. To solve the problems that the traditional target detection algorithm is slow, unable to realise real-time target detection in the underwater environment, the detection effect is poor, and the model structure is complicated and difficult to understand. This paper designs an embedded network model based on the advantages of the YOLOv5 network model. YOLO series is mainly designed for target detection network, because of its high real-time and accuracy is widely used in industry. The four variants of the YOLOv5 network are v5s, v5M, v5L, and v5x. After experimenting with the YOLOv5 family of networks versus YOLOv3-Efficientnet and YOLOv4-tiny, this paper chose YOLOv5s as the target detection model. Just a few of YOLOv5's numerous advantages include its excellent detection effect, strong practicability, small model size, cheap deployment cost, high degree of adaptability, and rapid detection speed. The advantages of its small model size stand out among the others. These advantages are especially suitable for embedded devices to complete the task of target detection. YOLOv5 is similar to previous versions, with Backbone, Neck, and Head. Use the dataset from this study as an illustration to describe how YOLOv5 is implemented. The backbone feature extraction network of YOLOv5 is CSPDarknet. The backbone extraction network CSPDarknet extracts feature information from sea cucumber and sea urchin photos after receiving them. The features extracted by CSPDarknet are called feature layers. The feature layer is a collection of features from the input sea cucumber and sea urchin images. Three feature layers are collected from the backbone section to build the subsequent network; these three feature layers are known as effective feature layers. The three useful feature layers that were obtained in the backbone part are then fused together. Combining feature data from various scales is the goal of feature fusion. The Head is responsible for evaluating the feature points and deciding whether or not there are objects that correspond to the feature points. The YOLOv5 model structure, as shown in Fig. 3.

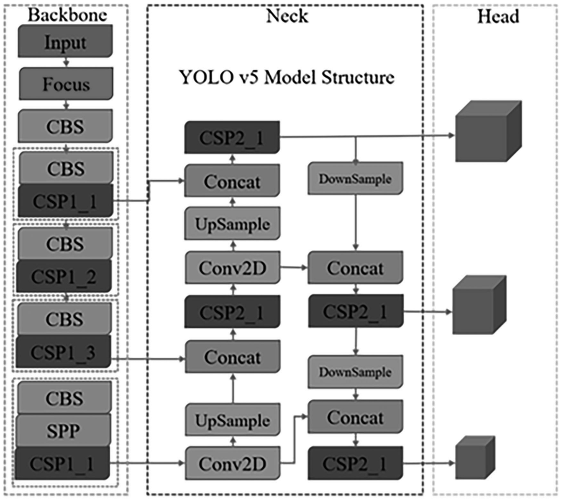

Figure 3. YOLO v5 model structure.

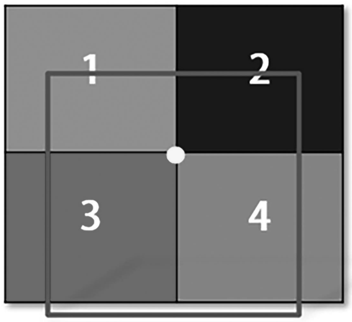

Figure 4. Mosaic stitching.

3.1.1 Input

To preserve the original benefits, the input of YOLOv5 utilises the Mosaic data enhancement method of YOLOv4. This part mainly takes the dataset of this paper as an example to describe. A new data augmentation technique called mosaic data augmentation combines four photos of sea urchins and sea cucumbers by randomly scaling, cropping, and arranging the individual images [24]. This method refers to CutMix and only mixes two images. The method to input the image is shown in Fig. 4.

It can be seen from Fig. 4 that the four colours represent the four images of sea cucumbers and sea urchins, and the excess parts will be discarded. The dataset can be improved by mosaic data. Four sea urchin and sea cucumber photos are randomly selected, resized, and dispersed before being utilised for splicing, considerably enhancing the detection dataset. This random scaling method adds a lot of small objects, improves the robustness of the network and reduces GPU memory. This method directly calculates the data of the four pictures, and setting a smaller mini-batch value can achieve better results.

Four original photographs, $X1$, $X2$, $X3$, and $X4$, shall be used. $X1$ shall be the image in the upper left corner,

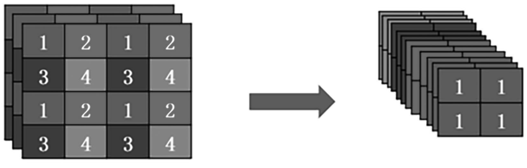

Figure 5. Alice operation.

$X2$ shall be the image in the upper right corner, $X3$ shall be the image in the lower left corner, and $X4$ shall be the image in the lower right corner. After splicing, Y is the outcome, and the size of the spliced picture is (H, W). Equation (3) then illustrates the mathematical formulation of the mosaic stitching technique.

Among them, Yh, w represents the pixel value of the spliced image at row h and column w. Through this formula, the four original pictures are stitched into a picture of size (H, W).

$$Y_{h,w} = \begin{cases} X_{1,h,w} & \text{if } 0 \le h < \frac{H}{2} \text{ and } 0 \le w < \frac{W}{2} \\ X_{2,h,w-\left(\frac{W}{2}\right)} & \text{if } 0 \le h < \frac{H}{2} \text{ and } \frac{W}{2} \le w < W \\ X_{3,h-\left(\frac{H}{2}\right),w} & \text{if } \frac{H}{2} \le h < H \text{ and } 0 \le w < \frac{W}{2} \\ X_{4,h-\left(\frac{H}{2}\right),w-\left(\frac{W}{2}\right)} & \text{if } \frac{H}{2} \le h < H \text{ and } \frac{W}{2} \le w < W \end{cases} \tag{3}$$

3.1.2 Backbone

Due to the limited computing power of embedded devices, the deployed model requires lightweight processing, and the Focus structure of YOLOv5 is a lightweight feature extraction module that can improve the efficiency and accuracy of the model. It can also downsample the features of the input image from high resolution to low resolution, and then upsample it from low resolution to high resolution, so as to obtain more comprehensive and rich feature information. Image extraction plays a very important role in solving the shortcomings of insufficient feature extraction. Prior to the picture entering the Backbone in YOLO v5, the Focus structure slices the image. Alice operation, as shown in Fig. 5.

3.1.3 Output

The YOLO series target identification loss function is composed of a classification loss function and a candidate box regression loss function. In recent years, the candidate box regression loss function has undergone continual improvement, and YOLO v5 employs the most current candidate box loss function, the CIOU loss function. The CIOU loss is calculated as follows:

$$\mathrm{L}_{CIoU} = 1 - IoU + \frac{P(b, b^{\mathrm{gt}})}{c^2} + \partial v \tag{4}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}} - \arctan\frac{w}{h}\right)^2 \tag{5}$$

$$IoU = \frac{|b \cap b^{\mathrm{gt}}|}{|b \cup b^{\mathrm{gt}}|} \tag{6}$$

$$\partial = \begin{cases} 0 & IoU < 0.5 \\ \frac{v}{(1-IoU)+v} & IoU \ge 0.5 \end{cases} \tag{7}$$

The prediction box and label box are denoted in the formula by the letters band and b^{gt}, respectively, while the width and height of the label box and prediction box are denoted by the letters w^{gt}, h^{gt}, w, and h. ∂ is the weight coefficient, and P is the distance between the two boxes' centre points.

3.2 Target Grasping Strategy

This paper presents a fuzzy PID controller to improve the success rate of grasping underwater objects using a single robotic arm. The controller receives target information, performs calculations, and outputs control commands to manipulate the movement of the ROV manipulator. The fuzzy PID algorithm is an effective control algorithm that enables the ROV robotic arm to accurately grasp the target object and move it to the specified position in the underwater environment. Moreover, the fuzzy PID algorithm can adjust the fuzzy set and fuzzy rules based on the actual underwater situation and adapt to the operating state of the ROV. In this study, three PID controllers are used for depth, heading, and grasping control, and the fuzzy control algorithm is added to achieve adaptability and robustness. To get the optimum control effect and system performance, the settings of the fuzzy PID controller are extensively tested and simulated underwater. Equation (8) gives the transfer function of the fuzzy PID controller in its general form. It should be noted that although this paper only uses a single robotic arm, the proposed fuzzy PID controller can be applied to other ROV systems with multiple robotic arms.

$$G(s) = Kp + \frac{Ki}{s} + \frac{Kd \ \cdot N}{1 + \frac{N}{s}} \tag{8}$$

Among them, Kp, Ki, and Kd are the gains of the proportional, integral, and differential controllers, respectively, and N is the fuzzy factor of the fuzzy controller. The output of the fuzzy PID controller can be expressed by (9).

$$y(t) = K_p \cdot e(t) + K_i \cdot \int e(t)\mathrm{d}t + K_{\mathrm{d}} \cdot N \cdot \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{9}$$

where $e(t)$, the difference between the reference input and the feedback input, represents the error signal.

The output of the fuzzy PID controller is the superposition of the output of the PID controller and the output of the fuzzy controller. After the motion control design of ROV is completed, solve the serial communication between STM32 and Raspberry Pi. Later, according to the PWM opening and closing signal value of the G30 robotic arm, the PWM signal output of the STM32 is designed.

4. Experiment and Analysis

This part analyses from the experimental platform construction, target detection experiment, and grasping experiment. Target detection and grasping will be described in detail in this section.

4.1 Experimental Platform

ROV is an underwater vehicle that can move freely underwater. It is equipped with cameras, thrusters, depth gauges, and various sensors to perceive the surrounding environment, and can also be equipped with robotic arm and other equipment to complete related tasks instead of humans. After the above software and hardware design and the establishment of the experimental environment. In this paper, the swimming pool is used to build an underwater environment according to the requirements, and ROV is selected as the tool to complete the entire target detection and grasping platform. Target detection and grasping experimental platform, as shown in Fig. 6.

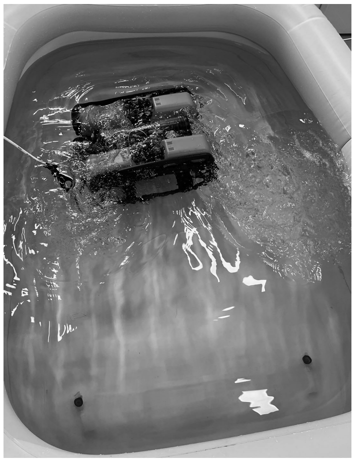

Figure 6. ROV target grasping strategy based on vision perception.

Table 1
Raspberry Pi 4B Configuration Parameters

Parameter	Value
SOC	Broadcom BCM2711
CPU	64 Bit 1.5 GHz Quad Core (28 nm process)
GPU	Broadcom Video Core VI@ 500 MHz
WIFI network	802.11AC Wireless 2.4 GHz/5 GHz

This paper combines the computing power of embedded devices with the existing server resources of the laboratory. The PC GPU selected in this article is GTX 3060, the memory of the graphics card is 8 GB, the processor used is AMD Ryzen 5 5500, and the machine has 32 GB of RAM. The training environment of the network model is configured with PyTorch 1.7.1, and the CUDA version is 11.0. Download and install dependent libraries according to network model training requirements.

In view of the current development of embedded devices, the embedded device selected in this article is the Raspberry Pi 4B. The system installed on Raspberry Pi 4B is Ubuntu18.04. The relevant technical parameters involved in this Raspberry Pi 4B, as shown in Table 1.

The target grasping task is mainly to choose a suitable manipulator. This paper chooses the G30 single-function manipulator. G30 Robotic Arm Configuration Parameters, as shown in Table 2.

4.2 Target Detection Experiment

4.2.1 Evaluation Metrics

Commonly used evaluation indicators for target detection algorithms include mAP, AP, $F1$-Score, IOU, NMS, FPS, *etc.* The experimental environment built in this paper is an underwater environment, and the target detection model is

Table 2
G30 Robotic Arm Configuration Parameters

Parameter	Value
PWM neutral signal	1500 μs
PWM open signal	>1530–1900 μs
PWM close signal	<1470–1100 μs
Time to open/close	1.6 s

deployed to embedded devices. In this paper, combined with the actual situation, the evaluation indicators of the target detection model are, mAP and FPS. FPS is the number of frames per second to process images, that is, how many images are processed per second. mAP is the average AP value, and the average AP value is calculated for multiple validation set individuals. mAP calculates the average area under the P-R curve across all categories, while AP determines the area under a specific sort of P-R curve. Precision can be expressed (10), recall can be expressed by (11), mAP can be expressed by (12), and (13) can be used.

$$P = \frac{\mathrm{TP}}{TP + FP} \tag{10}$$

$$R = \frac{\mathrm{TP}}{TP + FN} \tag{11}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N}\int_{0}^{1} P(R)\mathrm{dR} \tag{12}$$

$$F1 = 2\frac{\mathrm{PR}}{P + R} \tag{13}$$

P represents the number of real sea cucumber pictures among the recognised pictures. R represents the ratio of the number of correctly identified sea cucumbers to all real sea cucumbers in the test set. TP represents the number of sea cucumber pictures that are correctly detected, FP represents the number of sea cucumber pictures that are detected as sea cucumber pictures, and FN represents the number of sea cucumber pictures that are not detected, and the system mistakenly thinks it is sea urchin.

4.2.2 Model Effect Analysis

After the network model training is completed, select the sea cucumber pictures in the verification set for verification. The results in Fig. 7 show that YOLOv4-Tiny and YOLOv5s can identify sea cucumbers, but the YOLOv3-Efficientnet model has missed detection, which has certain limitations in the subsequent target grasping process. From the above analysis, the YOLOv4-Tiny and YOLOv5s models have certain applicability. This paper continues to analyse the mAP and $F1$ of the model to obtain the best network model.

The network model is only verified on the sea cucumber dataset, which cannot accurately represent the overall performance of the network model. Therefore, it is necessary to further analyse the real performance evaluation indicators before drawing conclusions. This paper also analyses the mAP of the three network models. The value of mAP can intuitively reflect the accuracy of the target detection of the network model. Therefore, it is authoritative to choose mAP as the evaluation index of the three network models. The mAP of the three network models of YOLOv3-Efficientnet, YOLOv4-Tiny, and YOLOv5s are shown in Fig. 8.

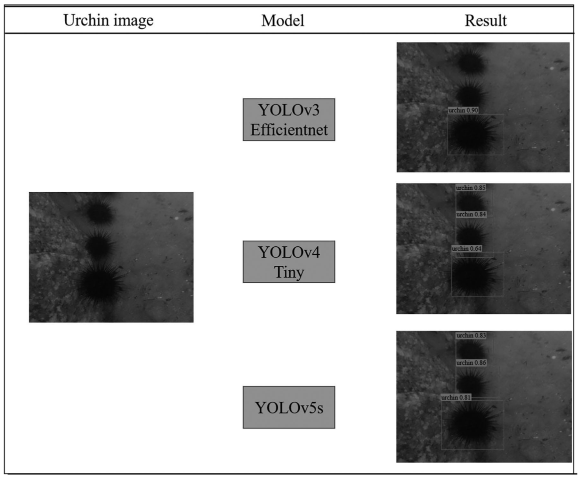

Figure 7. Network model validation.

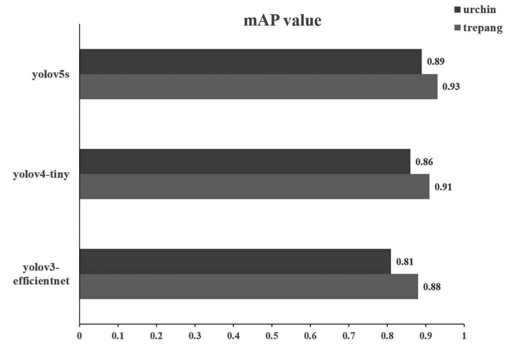

Figure 8. Network model mAP.

It can be seen from Fig. 8 that the mAPs of sea urchin and sea cucumber detected by the YOLOv5s network model are 0.89 and 0.93, respectively. Based on the above results, it can be seen that under the same experimental environment, the YOLOv5s network model is more suitable for identifying sea cucumbers and sea urchins. According to the analysis of real data, the three models have better recognition effects on sea cucumbers and higher accuracy, while the recognition accuracy of sea urchins will be relatively low. Sea cucumbers performed better than sea urchins for two possible reasons. The first is that the characteristics of sea cucumbers are more obvious, while the characteristics of sea urchins are more difficult to extract. The second is the interference of underwater environmental factors. The image of the sea urchin and the underwater environment are generally grey, which makes it difficult for the main feature extraction network to easily extract the features of the sea urchin, resulting in low detection accuracy.

According to the mAP study, YOLOv5s is more effective at detecting sea cucumbers and sea urchins as targets. Neither precision nor recall can be used as a comprehensive evaluation model, so the comprehensive evaluation index is selected to further compare the effectiveness of the three network models. To simplify the comparison steps, this paper only compares the three network models to select the effect of sea cucumber recognition, as shown in the Fig. 9.

In many cases, the model does not only care about a certain indicator but needs to balance the indicator value. The

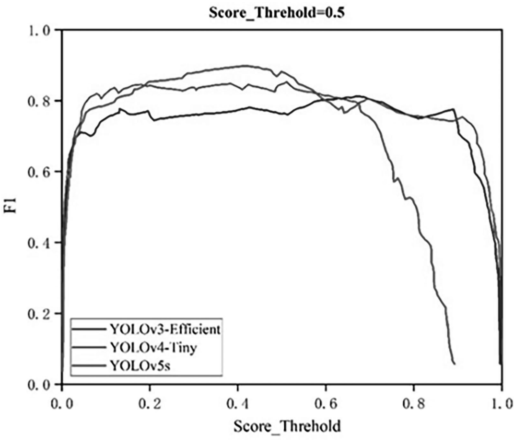

Figure 9. Network model $F1$ comparison.

Table 3
Model Performance

Model	mAP	FPS/s	Size/MB
YOLOv3-EfficientNet	0.84	4	25.6
YOLOv4-Tiny	0.88	10	24
YOLOv5s	0.91	8	14.7

$F1$ value comprehensively considers the accuracy and integrity of the model and is one of the important indicators to evaluate the performance of the target detection model. The confidence score in this study was set to 0.5. Based on experimental results, it was observed that YOLOv5s performed better than the other two network models. Considering the characteristics of the underwater environment, it can be concluded that YOLOv5s is more suitable as an embedded network model for the target detection module.

By analysing sea cucumber and sea urchin image recognition and mAP, as can be observed, this platform benefits more from the YOLOv5s target detection network approach. This research examines the performance of the three models in additional detail while taking into account the intricate grabbing environment of underwater sea cucumbers and sea urchins. The three network models' individual performance, as shown in Table 3.

The model sizes of YOLOv3-Efficientnet, YOLOv4-Tiny, and YOLOv5s are 25.6MB, 24MB, and 14.7MB, respectively. The FPS of the three network models is 4, 10, and 8, respectively. The size of the YOLOv3-Efficientnet and YOLOv4-Tiny models exceeds 20MB, and these two network models have no advantage over YOLOv5s. Due to the low computing power of embedded devices, when the network model is large, the accuracy of target detection is reduced when deployed to embedded devices. The two models cannot satisfy ROV for real-time target detection and grasping. The analysis of YOLOv4-Tiny and YOLOv5s shows that under the same experimental conditions, the FPS of YOLOv4-Tiny is higher than that of the YOLOv5s network model. Since this paper needs to perform real-time target detection and grasping of sea urchins, the real-time performance and accuracy need to be considered. Although the frame rate of YOLOv4-Tiny deployed on embedded devices is higher than that of the YOLOv5s network model, the mAP of the YOLOv5s network model is higher than that of YOLOv4-Tiny when the frame rate is guaranteed. Based on the above analysis, it can be seen that although the YOLOv5s model is not perfect, it has high accuracy and detection efficiency in sea cucumber and sea urchin image recognition, and can complete the real-time target detection and grasping tasks of ROV. In this paper, based on the actual situation and requirements of the platform, YOLOv5s is selected as the network model for the target detection task of the platform.

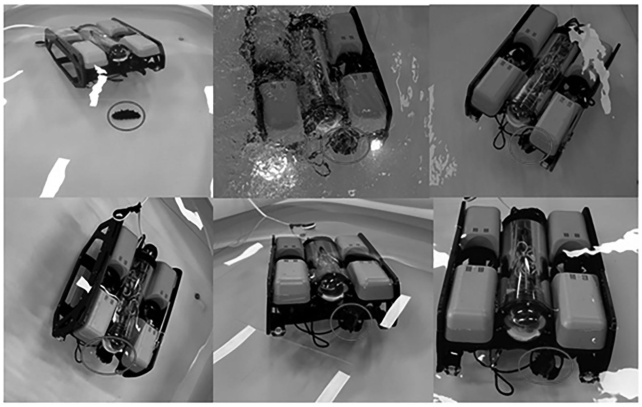

Figure 10. Underwater target grasping process.

4.3 Target Grasping Experiment

This section mainly analyses the results of target grasping and optimises the grasping strategy and target detection through the results.

4.3.1 Experiment and Analysis

In this paper, YOLOv5s is selected as the network model for the target detection task, so the target grasping analysis mainly discusses the grasping effect of the YOLOv5s model deployed on the ROV platform. The camera of ROV collects images of sea cucumbers and sea urchins, and the trained YOLOv5s target detection network processes them. After obtaining the detection results, they are sent to STM32. STM32 receives the data, sends grasping signals to control the robotic arm, and performs grasping tasks.

In this paper, the pool is used to simulate the underwater environment to perform the grasping task, and two underwater environments are set according to the requirements, namely, clear water quality and turbid water quality, and comparative experiments are carried out to observe the target grasping effect of this platform. The platform designed in this paper performs the process of grabbing sea cucumbers, as shown in Fig. 10.

Figure 10 shows the state of ROV performing the sea cucumber grasping task, from the initial preparation state to the whole process of ROV identifying and capturing sea cucumbers. From the red mark in the figure, it can be clearly seen that the ROV can perfectly realise the process of identifying and grasping the target sea cucumber.

4.3.2 Comparison of Different Algorithms

In this paper, fuzzy control and PID control are combined to design a controller. The effect of PID and fuzzy PID on the grasping of the robotic arm is compared by simulation. The specific results are shown in Table 4.

Table 4
Results of Different Algorithms

Algorithm	Maximum Deviation/cm	Steady State Error/cm	Response Time/s
PID	4	8	4
Fuzzy PID	3	5	2

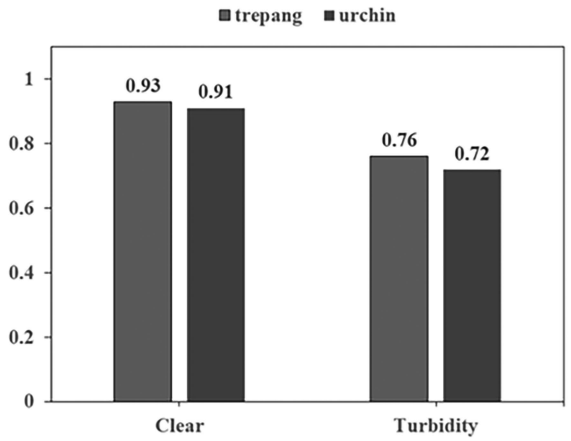

Figure 11. Sea cucumber and sea urchin grasping success rate.

To compare the impacts of the fuzzy controller and fuzzy PID controller. In this paper, the maximum deviation, steady-state error, and response time are selected as evaluation indicators. Among them, the maximum deviation, steady-state error, reaction time, and other indications favour the fuzzy PID controller over the fuzzy controller. In summary, the combination of fuzzy and PID can better complete complex and changeable underwater grasping tasks.

The simulation experiment can judge that the fuzzy PID is more suitable for the grasping of the robotic arm, and it has passed 200 underwater experiments, as shown in Fig. 11.

By analysing the average grasping success rate of the target detection and grasping strategy designed in this paper under different water quality environments, and then comparing the grasping success rates of sea urchins and sea cucumbers under the same water quality environment. This experiment mainly analyses the successful grasping rate of sea cucumber and sea urchin by the strategy proposed in this paper under the clear water quality and turbid water quality experimental environment. In the clear underwater experimental environment, the grasping success rates of sea cucumbers and sea urchins were 0.93 and 0.91, respectively. In the turbid underwater experimental environment, the grasping success rates of sea cucumbers and sea urchins were 0.76 and 0.72, respectively. According to the analysis of real experimental data, the average grasping rate of sea cucumbers is higher than that of sea urchins, which may be due to the different shapes and sizes of sea cucumbers and sea urchins. The shape of the sea cucumber is cylindrical, while the shape of the sea urchin is spherical. The opening of the gripper of the G30 robotic arm is 70 mm. The shape and size of the sea cucumber are more suitable for the grasping of the robotic arm. Therefore, the success rate of sea cucumbers is slightly higher than that of sea urchins.

5. Conclusion

To solve the problems of low fishing efficiency and high fishing risk coefficient in complex underwater environment, an ROV target grasping strategy based on visual perception is proposed. The experimental results prove that the YOLOv5s target detection network model can better realise target recognition, and the capture success rate after combining fuzzy and PID is higher. In this paper, different experimental environments are set for grasping experiments, and the YOLOv5s target detection network model is selected to ensure the detection accuracy and speed. In a clear underwater environment, the average grasping success rate is 0.92. Under turbid water quality, the average grasping success rate is 0.74. In summary, the target detection and grasping strategy based on underwater visual perception not only ensures a lightweight network model but also improves the detection accuracy and realises the task of real-time fishing, which has practical application value for aquaculture and fishing. It is true that the target detection model involved in this paper still has certain shortcomings. In the follow-up research, it is necessary to use a more flexible loss function to further strengthen the learning ability of each target detection sample so as to improve the convergence speed of the target detection model. In practical applications, in the face of complex environments, the performance of the model is improved through adaptive data augmentation.

Acknowledgement

The Natural Science Foundation of Anhui Province under Grant 2308085MF204, the University Industry-University Research Innovation Fund of China under Grant 2021ZYA11013, the University Synergy Innovation programme of Anhui Province under Grant GXXT-2021-010, and the National Natural Science Foundation of China under Grant 61773177 all contributed to the funding of this work.

References

[1] S. Cui, Y. Wang, and S. Wang, Real-time perception and positioning for creature picking of an underwater vehicle, *IEEE Transactions on Vehicular Technology, 69*(4), 2020, 3783–3792.

[2] Z. Guan, C. Hou, S. Zhou, and Z. Guo, Research on underwater target recognition technology based on neural network, *Wireless Communications and Mobile Computing*, *2022*, 2022, 4197178.

[3] N. Kapetanović and K. Krčmar, Tether management system for autonomous inspection missions in mariculture using an ASV and an ROV, *IFAC-PapersOnLine, 55*(31), 2022, 327–332.

[4] Y. Zhao, Y. Shi, and Z. Wang, The improved YOLOv5 algorithm and its application in small target detection, *Proc. International Conf. on Intelligent Robotics and Applications, Cham*, 2022, 679–688.

[5] J. Yun, D. Jiang, Y. Liu, and Y. Sun, Real-time target detection method based on lightweight convolutional neural network, *Frontiers in Bioengineering and Biotechnology*, *10*, 2022, 861286.

[6] K. Hu, J. Jin, and F. Zheng, Overview of behavior recognition based on deep learning, *Artificial Intelligence Review*, *56*, 2023, 1–33.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, *Proc. 2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, 580–587.

[8] J. Redmon, S. Divvala, and R. Girshick, You only look once: Unified, real-time object detection, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, 779–788.

[9] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks,

Advances in Neural Information Processing Systems, 9199, 2015, 2969239–2969250.

[10] J. Redmon and A Farhadi, YOLO9000: Better, faster, stronger, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, 7263–7271.

[11] W. Liu, D. Anguelov, and D. Erhan, SSD: Single shot multibox detector, *Proc. European Conf. on Computer Vision, Cham*, 2016, 21–37.

[12] X. Wang, X. Hua, and F. Xiao, Multi-object detection in traffic scenes based on improved SSD, *Electronics, 7*(11), 2018, 302.

[13] F. Han, J. Yao, and H. Zhu, Underwater image processing and object detection based on deep CNN method, *Journal of Sensors*, *2020*, 2020, 6707328.

[14] D. Jiang, G. Li, and Y. Sun, Manipulator grabbing position detection with information fusion of color image and depth image using deep learning, *Journal of Ambient Intelligence and Humanized Computing, 12*(12), 2021, 10809–10822.

[15] H. Sun, X. Cui, and Z. Song, Precise grabbing of overlapping objects system based on end-to-end deep neural network, *Computer Communications, 176*, 2021, 138–145.

[16] F. Mei and X. Gao, Target recognition and grabbing positioning method based on convolutional neural network, *Mathematical Problems in Engineering*, *2022*, 2022, 4360346.

[17] G.U. Shaokui, and L.I. Longyan , Research status of robot grab detection based on vision, *Asian Journal of Research in Computer Science, 14*(4), 2022, 21–35.

[18] L. Trottier, P. Giguere, and B. Chaib-Draa, Convolutional residual network for grasp localization, *Proc. 2017 14th Conf. on Computer and Robot Vision (CRV),* Edmonton, AB, 2017, 168–175.

[19] T.H. Nguyen, T.T. Nguyen, and T.V. Tran, A method for localizing and grasping objects in a picking robot system using kinect camera, *Proc. International Conf. on Intelligent Human Computer Interaction, Cham*, 2021, 21–26.

[20] M.W. Walker, Manipulator kinematics and the epsilon algebra, *IEEE Journal on Robotics and Automation, 4*(2), 1988, 186–192.

[21] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods, *Proc. 2018 IEEE International Conf.on Robotics and Automation (ICRA),* Brisbane, QLD, 2018, 6284–6291.

[22] A. Jebelli, H. Chaoui, A. Mahabadi, and B. Dhillon, Tracking and mapping system for an underwater vehicle in real position using sonar system, *International Journal of Robotics and Automation, 37*, 2022, 124–134.

[23] Z. Geng, Study on the position control of electric cylinder based on fuzzy adaptive PID, *International Journal of Robotics and Automation, 35*(3), 2020, 242–247.

[24] C. Li, H. Gao, and Y. Yang, Segmentation method of high-resolution remote sensing image for fast target recognition, *International Journal of Robotics and Automation, 34*(3), 2019, 4597–4618.

Biographies

Jiawen Li received the B.S. degree in software engineering from Chuzhou University, Chuzhou, China, in 2021. He is currently pursuing the M.S. degree with Anhui University, Hefei, China. His research interest includes underwater robot vision.

Xiang Cao was born in Sichuan, China. He received the B.Sc. degree in electronic and information engineering from Southwest University, Chongqing, China, in 2004, the M.Sc. degree in communication and information systems and the Ph.D. degree in power electronics and power transmission from Shanghai Maritime University, Shanghai, China, in 2011 and 2016, respectively. Since 2016, he is doing postdoctoral research with Southeast University. He is currently an Associate Professor with the School of Artificial Intelligence, Anhui University. His current research interests include target searching and path planning of underwater vehicles.

Xueyou Huang received the B.S. degree in computer science and technology from the Chengdu University of Information Technology, Chengdu, China, in 2021. He is currently pursuing the M.S. degree with Anhui University, Hefei, China. His research interests include object detection and visual object tracking.