IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 July 2023, accepted 11 August 2023, date of publication 15 August 2023, date of current version 21 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305339

APPLIED RESEARCH

An Object Recognition Grasping Approach Using Proximal Policy Optimization With YOLOv5

QINGCHUN ZHENG[1], ZHI PENG[2], (Graduate Student Member, IEEE), PEIHAO ZHU[1], YANGYANG ZHAO[3], (Graduate Student Member, IEEE), RAN ZHAI[2], AND WENPENG MA[4]

[1]Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, School of Mechanical Engineering, Tianjin University of Technology, Tianjin 300384, China
[2]School of Mechanical Engineering, Tianjin University of Technology, Tianjin 300384, China
[3]School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
[4]National Demonstration Center for Experimental Mechanical and Electrical Engineering Education, Tianjin University of Technology, Tianjin 300384, China

Corresponding author: Peihao Zhu (zhupeihao_ch@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 62073239.

ABSTRACT Aiming at the problems of traditional grasping methods for mobile manipulators, such as single application scenarios, low accuracy, and complex grasping tasks, this paper proposes an object recognition grasping approach using Proximal Policy Optimization (PPO) with You Only Look Once v5 (YOLOv5), which combines a vision recognition algorithm with a deep reinforcement learning algorithm to achieve object recognition grasping. First, YOLOv5 is adopted to identify the object and obtain the location information. Second, the PPO algorithm is used for object grasping to obtain the grasping strategy. Third, the PPO algorithm is compared with the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms in batches 16 and 128, respectively. The average reward training results of the PPO, SAC, and TRPO algorithms are obtained in our work. Experimental results show that the proposed method, in terms of object recognition speed, outperforms the original YOLOv4 model. The YOLOv5 model achieves 96% precision on our own built recognition dataset, which has higher detection precision and lower hardware requirements than the YOLOv4 model. Our proposed method outperforms SAC and TRPO algorithms in object grasping, and the average reward of the PPO algorithm is improved by 93.3% and 41% compared to SAC and TRPO algorithms, respectively. Finally, through the comparison of ablation experiments, our method has the highest accuracy and mean average precision (mAP)@0.5 value of 92.3%. We demonstrate in actual physical experiments that the grasping success rate under our proposed approach reaches 100%, providing a new research strategy for object grasping by the robot manipulator.

INDEX TERMS Deep reinforcement learning, manipulator, object grasping, proximal policy optimization, YOLOv5.

I. INTRODUCTION

The global manufacturing industry has led to the widespread use of mobile manipulators (MM) in industrial transport and agricultural picking. To increase the range of the manipulator, a mobile base is combined with the manipulator to form a mobile manipulator. The mobile manipulator combines the mobility of a mobile base with the flexible operation capabilities of a manipulator [1]. This technology can be utilized in hazardous environments or for handling dangerous objects, ultimately reducing the risk of harm to humans [2]. Additionally, the mobile manipulator is capable of mimicking human behavior when it comes to object grasping [3]. However, identifying objects with complex surfaces remains a challenge in the field of classical image processing [4].

Due to the various sizes and shapes of objects, a sizable number of datasets must be trained to overcome the hurdles of grasping and recognition in complex object environments. In recent years, mobile manipulators have been widely utilized for grasping, but they still have problems such

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

as a single working mode, sensitivity to the surroundings, incorrect location, and poor recognition performance [5]. Before grabbing an object, the mobile manipulator must first determine the object's position and increase identification accuracy. Therefore, the object recognition and object grasping technology of robot manipulators are worth studying.

Deep learning-based target recognition and reinforcement learning-based grasping have emerged as two crucial mobile manipulator technologies for mobile applications such as industrial handling and agricultural picking. However, object detection and gripping accuracy need to be further improved in complex circumstances. Target recognition algorithms are widely employed in a variety of fields, including transportation, industry, and agriculture. The robust autonomous learning capabilities of deep learning-based recognition algorithms can help mobile manipulators detect and identify items more precisely in complex environments. Object recognition is being researched for the mobile manipulator grasping job to better localize and recognize items so that the robot can grab them more precisely [6]. The continual interaction between the agent and the environment allows the reinforcement learning-based grasping algorithm to maximize reward, and continuous training and learning are intended to increase the object grasping accuracy.

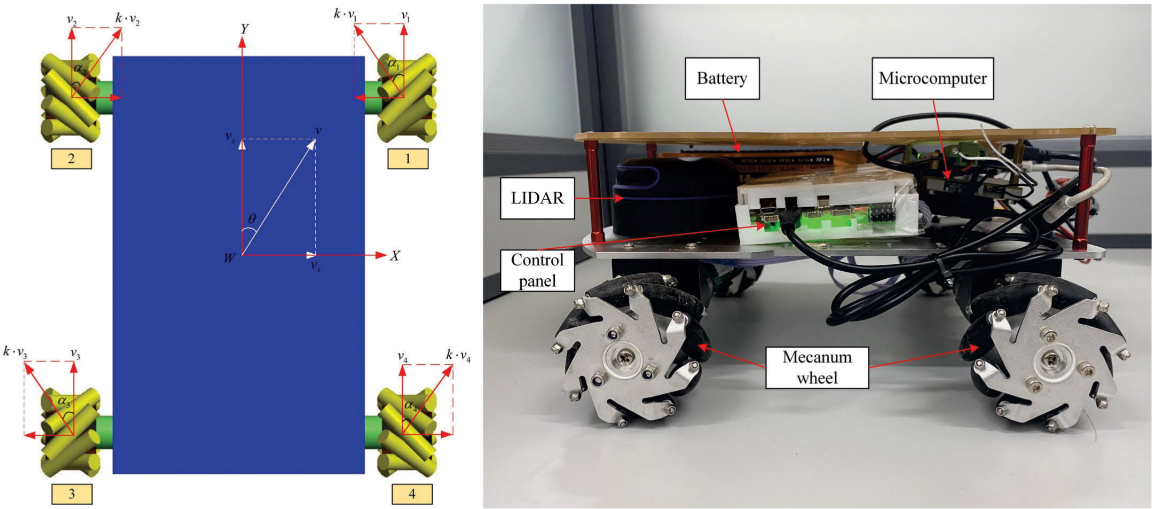
At present, object recognition algorithms, including a one-stage and a two-stage object detection algorithm, are primarily based on deep learning. The one-stage object detection algorithm mainly includes Single Shot MultiBox Detector (SSD), You Only Look Once (YOLO), and YOLOv2∼v8. The two-stage object detection algorithm mainly includes Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, and Faster R-CNN. Compared to the two-stage object detection technique, the one-stage approach detects objects more quickly. Additionally, YOLOv5, as one of the classical algorithms in the one-stage object detection algorithm, can simplify the network structure and optimize the activation function. More significantly, the YOLOv5 network employs a straightforward Convolutional Neural Network (CNN), in contrast to previous networks. YOLOv5 is faster than other networks and is better suited for real-time target identification applications because of its smaller and simpler model. A Mask-SSD model was proposed in [7] to improve the detection performance of small targets. The authors in [8] proposed an object recognition algorithm with a sparse detection algorithm to modify YOLOv5, which aims to build a smaller model with a lower cost to improve the detection efficiency of small objects. In [9], the authors proposed a novel object detection method based on a wireframe feature, which used Fast R-CNN to detect objects in which the mean average precision (mAP) is 89.4%.

Next, let us consider the case of the robot manipulator with grasping. In addition to the need to improve the object recognition ability, the accurate grasping ability of the mobile manipulator in complex environments also needs further research. Currently, grasping algorithms are mainly based on traditional methods and deep reinforcement learning (DRL) algorithms. A novel robot rigid object pickup method was proposed in [10] to grasp objects mixed with towels without using a specific posture detection method, which is not suitable for flexible hybrid scenes. In [11], the authors proposed a dynamic evaluation method that extends the static evaluation method for grasping rigid objects to deformed objects, which is expected to provide a new research idea for grasping deformable objects. A grasping system based on deep reinforcement learning (DRL) with an improved soft actor-critic algorithm was proposed in [12] to speed up the learning process so that it can decouple object detection from deep reinforcement learning (DRL) control. The authors in [13] proposed a graph-based Q-learning model to effectively explore invisible objects and improve collaborative grasping performance, which can help robots grasp completely occluded objects in a cluttered scene.

Traditional grasping techniques only have one application situation and are incapable of producing acceptable results. However, one of the crucial jobs for robots is grasping in complex environments. As a result, using object grasping based on reinforcement learning to determine the location of target objects is effective. Further research is being done on grasping and object recognition for mobile manipulators. For instance, in the paper [14], they proposed a fast detection and grasping method based on improved Faster R-CNN, which can effectively complete detection and successfully grasp the object on the shelf. In [15], the authors investigated an intelligent mobile garbage collection robot based on visual recognition technology such that the robot detects and categorizes targets using the MobileNetv3-SSD deep learning algorithm, which also controls the manipulator to perform the garbage grasping task. A method based on YOLOv5 was proposed in [16] to detect objects and a method based on a deep deterministic policy gradient to grasp autonomous objects, which can be applied to robot arms with multiple degrees of freedom such that the simulation results are superior to the traditional methods.

The research mentioned above demonstrates that effective results can be obtained by combining a vision algorithm based on deep learning with a grasping strategy based on reinforcement learning. Less research is being done, nevertheless, in the area of robot applications. Therefore, this study suggests an object recognition grasping approach using Proximal Policy Optimization (PPO) with YOLOv5 to investigate the object recognition grasping of mobile robotic arms in more detail. The main contributions of this paper can be described as follows:

1) We propose an object recognition grasping approach based on Proximal Policy Optimization (PPO) with YOLOv5, which integrates the YOLOv5 algorithm and PPO reinforcement learning algorithm to achieve object recognition grasping, providing a new research idea for object recognition grasping.
2) The final experimental results show that our proposed method improves the average reward by 93.3% and



(a) (b)

FIGURE 1. The mobile robot experimental equipment of the actual physical manipulator is displayed via a mobile base model. (a) Mobile base coordinate system (for kinematic analysis); (b) Mobile base (for providing power).

41% at batch 128 compared to the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms, respectively. By ablation experiment comparison, our method has the highest accuracy and mean average precision (mAP)@0.5 values of 96% and 92.3%, respectively. In the physical experiments, we verified that the success rate of object grasping under our proposed scheme is 100%.

The structure of this paper is as follows. Section II presents the kinematics model of the mobile manipulator, namely the inverse kinematic and forward kinematic analyses. Section III introduces the main algorithms for object recognition and grasping. Section IV depicts the experiments and results. Section V provides the discussion and the limitations of the methods. In Section VI, we summarize the thesis and look forward to the next step.

II. KINEMATICS MODELING AND ANALYSIS

A. INVERSE KINEMATICS ANALYSIS

Two A-type and two B-type Mecanum wheels make up the mobile base of the mobile manipulator. The Mecanum wheel, which has flexible motion properties and is simple to use to create omnidirectional motion, is frequently used [17]. The mobile base model is shown in Fig. 1, which is used to present the mobile robot experimental equipment of the actual physical manipulator. Fig. 1 (a) shows the coordinate relationship of the mobile base, which is used as a reference coordinate system for kinematic analysis. Fig. 1 (b) shows the physical diagram of the mobile base, which is used to provide power to the mobile manipulator. To gather environmental data, the mobile base's front light detection and ranging (LIDAR) system is used. Mecanum wheels [18] can be utilized when an omnidirectional mobile vehicle is required. The vehicle can freely revolve about the center in addition to moving along the anticipated course. The running speed of the vehicle is set to 0.1 m/s.

The positive Y-axis direction represents the mobile manipulator's forward motion, and the positive X-axis direction represents the mobile manipulator's right motion. After defining the Y-axis and X-axis positive directions as such, the motion of the center point of the mobile base is decomposed as follow:

$$\left(v_x, v_y, w\right) \tag{1}$$

$$v_x = v \cdot \sin\theta \tag{2}$$

$$v_y = v \cdot \cos\theta \tag{3}$$

where v_x represents the velocity of the central point along the direction X-axis, v_y represents the velocity of the central point along the direction Y-axis, and w represents the angular velocity rotating around the central point.

Wheels 1 and 3 and Wheels 2 and 4 move in the same manner as Fig. 1 (a), respectively. The first wheel will advance and then turn to the left. The forward motion speed of the wheel is v_1. The forward motion velocity and lateral movement velocity are combined as $k \cdot v_1$.

$$v_i + k \cdot v_i \cdot \cos\alpha_i = v_y + a_i \cdot w,\ \ i = \{1, 2, 3, 4\} \tag{4}$$

$$k \cdot v_i \cdot \sin\theta = v_x + b_i \cdot w,\ \ i = \{1, 2, 3, 4\} \tag{5}$$

where i represents the wheel number and $i = \{1, 2, 3, 4\}$, k is the scale factor. a and b are the dimensional lengths of the mobile base. α represents the angle between forward velocity and resultant velocity. The form of the variation in (4) and (5) is shown in (6) and (7).

$$\tan\alpha_i = \frac{k \cdot v_i .\ \sin\alpha_i}{k \cdot v_i \cdot \cos\alpha_i} = \frac{v_x + b_i \cdot w}{-v_i + v_y + a_i \cdot w} \tag{6}$$

$$v_i = v_y + a_i \cdot w - \frac{v_x + b_i \cdot w}{\tan\alpha_i} \tag{7}$$

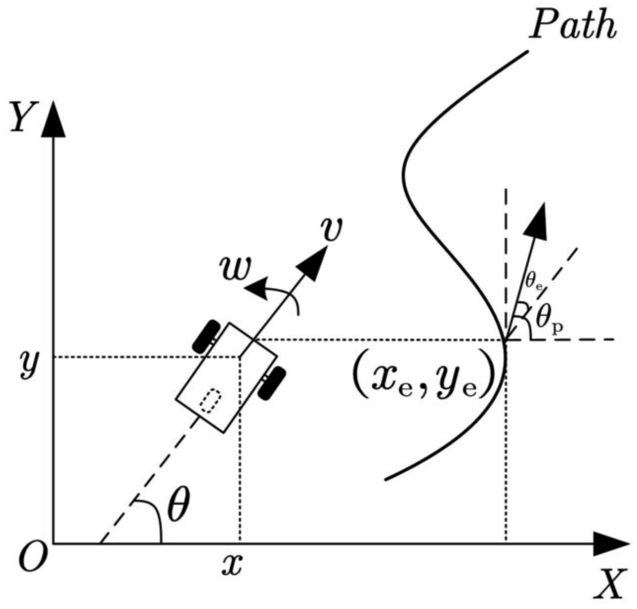


FIGURE 2. Path planning schematic of the mobile manipulator. (All variables are marked with black symbols, where the thick black arc represents the path.)

The lateral angle of the mobile base is 45°, where $a_i \in \{a, -a, -a, a\}$, $b_i \in \{b, -b, -b, b\}$, $\alpha_i \in \left\{\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}\right\}$, $\tan\alpha_i \in \{1, -1, 1, -1\}$.

$$\begin{cases} v_1 = v_y - v_x + a \cdot w + b \cdot w \\ v_2 = v_y + v_x - a \cdot w - b \cdot w \\ v_3 = v_y - v_x - a \cdot w - b \cdot w \\ v_4 = v_y + v_x + a \cdot w + b \cdot w \end{cases} \tag{8}$$

The motion velocity of the wheels is w_i, so $v_i = w_i \cdot R$. The relation between the rotation speed of the four wheels of the mobile base and the motion state of the center point is shown in (9).

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ -1 & 1 & -(a+b) \\ 1 & 1 & (a+b) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix} \tag{9}$$

where R represents the radius of the mobile base wheel. When the mobile base dimensions a, b and the wheel radius R are known, the rotation speed of each wheel can be calculated through the motion state of the center point.

B. FORWARD KINEMATICS ANALYSIS

As shown in Fig. 2, We define the notation $X = [x, y, \theta]^T$ as the actual posture of the mobile manipulator, (x, y) represents the actual position of the mobile manipulator, and θ represents the actual angle of the mobile manipulator. $u = [v, w]^T$ is the actual input of the system, v is the actual linear velocity of the mobile manipulator, and w is the actual angular velocity of the mobile manipulator. $[x_e, y_e, \theta_e]$ denotes the error vector, (x_e, y_e) represents the deviation between the actual position and the reference position, and θ_e is the angle deviation.

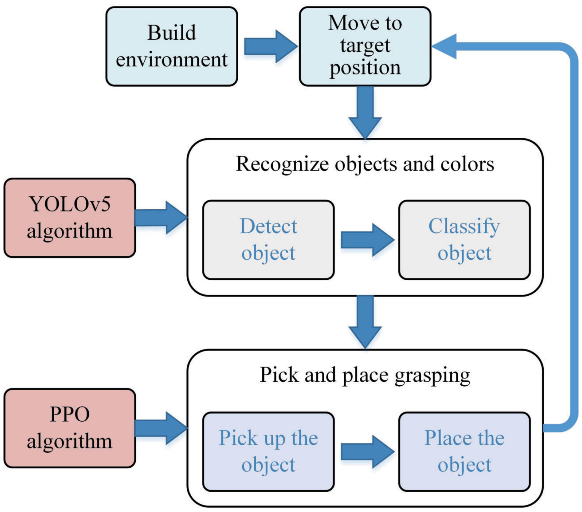


FIGURE 3. The interaction structure of the vision algorithm and grasp algorithm where the mobile manipulator interacts with the objects.

The kinematic model of the mobile manipulator can be described as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} \tag{11}$$

where $X_r = [x_r, y_r, \theta_r]^T$ represents the reference posture of the mobile manipulator, (x_r, y_r) represents the reference position of the mobile manipulator, and θ_r represents the reference angle of the mobile manipulator. $u_r = [v_r, w_r]^T$ represents the reference input of the system, v_r is the reference linear velocity of the mobile manipulator, and w_r is the reference angular velocity of the mobile manipulator.

The error model of the mobile manipulator is described as follows:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} [X_r - X] \tag{12}$$

III. MAIN ALGORITHMS FOR OBJECT RECOGNITION GRASPING

A. VISION ALGORITHM

Mobile manipulator control systems require highly accurate object detection. The visual-based recognition camera is frequently referred to as the manipulator's eye when it works in combination with manipulation [19]. The target's position must be ascertained and communicated to the mobile manipulator before it can grasp it. Fig. 3 presents the interaction structure of the vision algorithm and the grasping

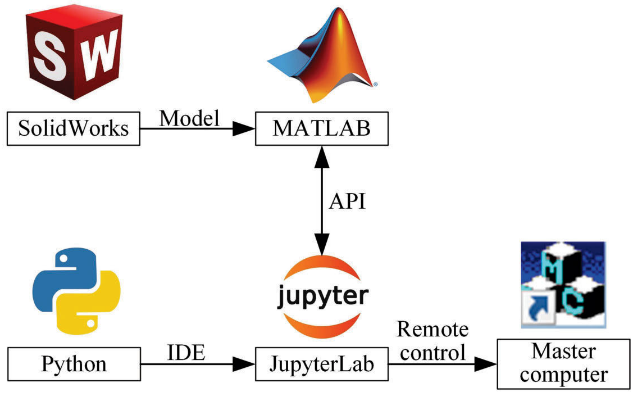


FIGURE 4. The interaction process of application programming interface (API) for mobile manipulators.

algorithm, where the mobile manipulator interacts with the objects. The environment is created, and the mobile base is brought to the desired location. The YOLOv5 vision algorithm is used in this study to identify colors and objects.

The DOFBOT arm and Jetson Nano [20] kit from NVIDIA, which have a robust artificial intelligence (AI) function and complete application programming interface (API) programmability, are used to construct the mobile manipulator independently in this research. A comprehensive desktop Linux environment with TensorFlow, PyTorch, Keras, Open Source Computer Vision Library (OpenCV), and Robot Operating System (ROS) is offered by the Jetson Nano, which runs Ubuntu 18.04.

The application programming interface (API) of the DOFBOT manipulator cannot be utilized right away. As depicted in Fig. 4, we employ Python for serial communication. Through a USB interface, we link the manipulator and Jetson board. The simulation experiment is carried out in MATLAB, and the mobile manipulator model is created in SolidWorks. A flexible building module for interactive computing is offered by JupyterLab [21]. JupyterLab, which focuses on interactive and exploratory computation, is interacted with using MATLAB. Although JupyterLab has a lot of functionality found in a conventional integrated development environment (IDE). The mobile manipulator is remotely controlled by the master computer using Jupyter debugging code while being simulated in real-time by the computer.

B. DATA COLLECTION

The major components of YOLOv5 are YOLOv5x, YOLOv5l, YOLOv5s, and YOLOv5m [22]. A lighter and more efficient YOLOv5s weight model is used to train the dataset. The YOLOv5, which stands for YOLOv5s, is used in later investigations. The accuracy of object recognition is increased by training a sizable dataset. The batch size is set to 16, the iteration epochs of the dataset are set to 50, and the image input size is 416 × 416 × 3.

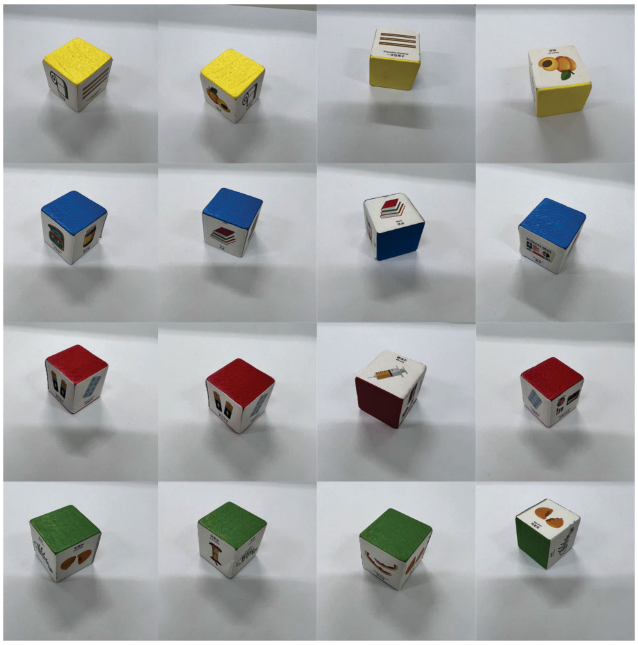


FIGURE 5. By collecting thousands of block images to build our recognition dataset, only a part of which is shown here. (Top: yellow block dataset; Second row: blue block dataset; Third row: red block dataset; Bottom: green block dataset.)

In this section, we create our YOLOv5 recognition dataset. As seen in Fig. 5, this study created a recognition dataset using the YOLOv5 vision algorithm. Thousands of images of yellow, blue, red, and green were taken in the real world to gather the images, and we have displayed a selection of them here to depict the four various orientations. However, overfitting may happen throughout the training and learning processes because of an insufficient amount of training data. In the future, we will expand datasets and introduce noise to reduce overfitting and enhance training. The number of crucial features to be extracted grows exponentially with the amount of training data.

C. REINFORCEMENT LEARNING ALGORITHM

Following object and color recognition by the mobile manipulator, we pick and position the object using reinforcement learning techniques. The mobile manipulator delivers a command to grip an object when the YOLOv5 vision algorithm detects it. In this study, we achieve the gripping strategy using the Proximal Policy Optimization (PPO) reinforcement learning method.

Deep reinforcement learning (DRL) has advanced significantly and attracted increasing attention from academia and industry as a result of the merger of deep learning (DL) and reinforcement learning (RL) [23]. A staged optimization decision forms the basis of the machine learning (ML) paradigm known as reinforcement learning (RL) [24]. A key component of the reinforcement learning framework is an agent that operates in a specific environment, allowing the agents to interact with it. The goal of the learning process is to maximize the reward, that is, to maximize the sum of rewards at all times.

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{13}$$

where R_t represents the reward function, γ represents the discount rate and $\gamma \in [0, 1]$. r is the reward, depending on the actions taken by the agent at the moment $t + k + 1$.

Soft Actor-Critic (SAC), Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO) are examples of common model-free reinforcement learning algorithms. The SAC algorithm can effectively tackle the problem of each gradient step requiring a large number of new samples [25]. A reinforcement learning technique that is both online and non-policy is the Soft Actor-Critic (SAC) algorithm [26]. Trust Region Policy Optimization (TRPO) alternates between updating policy parameters by resolving constraint optimization problems and sampling data interactively through the environment [27]. By keeping the revised policy within the trust region close to the present policy, the TRPO algorithm prevents a significant performance reduction as compared to the traditional policy gradient method. However, TRPO computation is difficult and necessitates numerous environmental interactions. The computation can be made simpler and more affordable by using the Proximal Policy Optimization (PPO) algorithm. By substituting the penalty term with the KL divergence and utilizing a clip-based truncation operation to decrease the update magnitude of the policy, PPO streamlines the TRPO method [28]. The actor-critic (AC) architecture used to develop the PPO algorithm increases its convergence speed compared to TRPO [29]. To grasp objects, this paper employs the PPO algorithm.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENT ENVIRONMENT

Python and MATLAB 2022a are used in the simulation experiment to create the simulation environment. Using Robot Operating System (ROS) nodes and a remote application programming interface (API) [30], a distributed control structure is realized. Python and an API are used to run the interactive simulation. Additionally, we built a server for the experimental environment with an Intel(R) Core i7-11800H CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU. The operating system is Ubuntu 18.04 with OpenCV 4.1 and TensorFlow 1.4.

We set the algorithm's maximum episode count at 50, the maximum episode length at 50, and the average episode length at 30 to finish the object-grasping job and increase the grasping success rate. We set the discount factor at 0.99 and the learning rates of the actor and critic at 0.001 and 0.001, respectively.

B. OBJECT RECOGNITION RESULTS

The mobile manipulator used in this study has a high-definition camera that can record both color and object images. The photos are tagged using LabelImg, and the trained YOLOv5 neural network model is used for training [31]. JupyterLab software analyzes the visual data captured by the camera, identifying objects and colors.

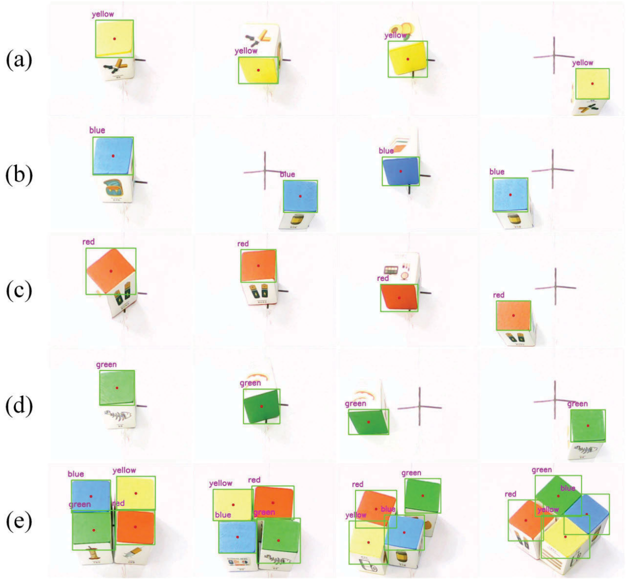


FIGURE 6. Results display for color detection. (a) Yellow color; (b) Blue color; (c) Red color; (d) Green color; (e) Four blocks colors. (Color detection is marked with a green box and purple text, respectively.)

We must calibrate objects and adjust colors by adjusting the Hue-Saturation-Value (HSV) threshold before we can recognize color, an operation that is automatically updated in real-time. To eliminate the distracting colors, the HSV [32] color model's high and low thresholds are modified. The hexcone model, often known as the intuitive properties of color, served as the basis for the creation of the color space known as HSV. The results of color recognition are displayed in Fig. 6. The colors of the yellow, blue, red, and green blocks may be distinguished from four distinct angles. The results demonstrate that the four color blocks can be precisely identified by the vision algorithm.

The results of object recognition are shown in Fig. 7. On each of the four faces of each color block, there are four unique things. The objects on each face are recognized, and Table 1 displays the particular confidence score [33]. In Table 1, group (a) objects have a high confidence score of 96%. Objects in group (b) with a minimum confidence score of 95%. The minimum confidence rating of 94% for group (c) objects. The identification area is represented by the cross in Fig. 7. When we position each object in turn, the recognition area will soon be recognized by the camera that is attached to the recognition program.

The confidence curve for the results of object recognition, which corresponds to the label classification in Table 1, is shown in Fig. 8. Toilet paper has remained constant at 0.96, as seen in Fig. 8 (a), while disposable chopsticks gradually declined to reach the lowest confidence score at

TABLE 1. The confidence scores of the four groups of object recognition results correspond to the objects in Fig. 7.

Groups	Objects	Confidence	Groups	Objects	Confidence
(a)	Toilet paper	0.96	(c)	Battery	0.96
	Disposable chopsticks	0.96		Pill	0.94
	Cigarette butts	0.96		Expired cosmetics	0.96
	Peach pit	0.96		Syringe	0.96
(b)	Zip-top can	0.96	(d)	Apple core	0.96
	Newspaper	0.96		Watermelon rind	0.96
	Book	0.95		Egg shell	0.92
	School bag	0.95		Fish bone	0.94

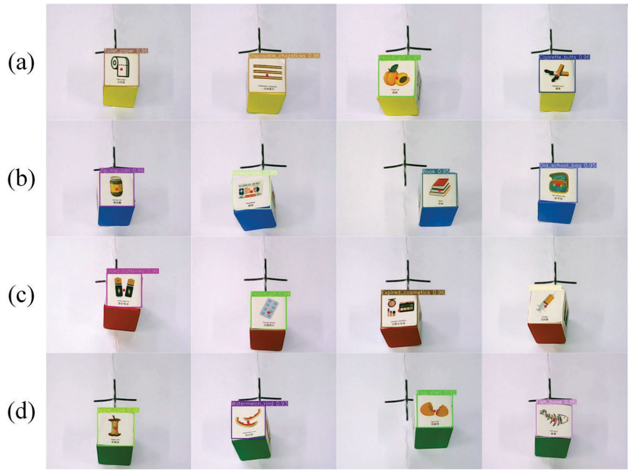
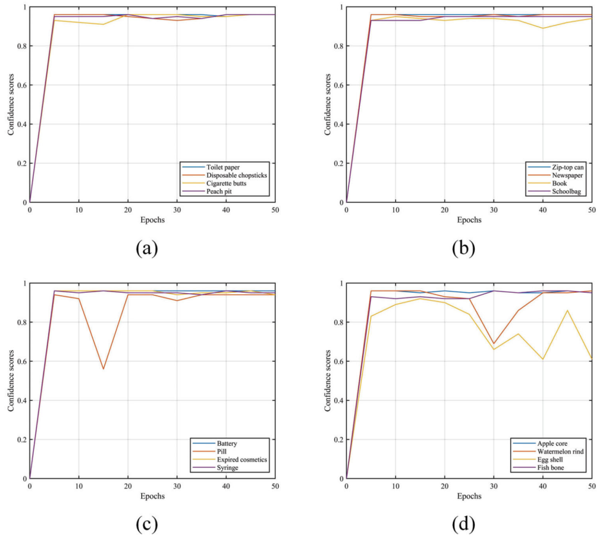
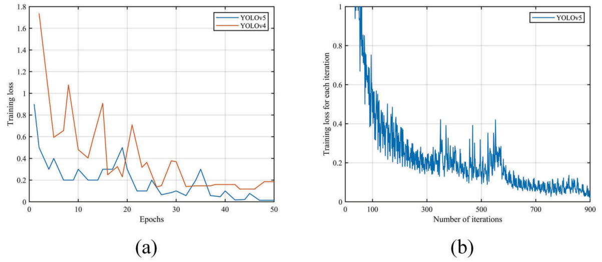


FIGURE 7. Results display for object detection. (a) Yellow block objects; (b) Blue block objects; (c) Red block objects; (d) Green block objects. (Object detection is marked with different color boxes and text, respectively.)



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

FIGURE 8. The confidence curves of object recognition results corresponds to the label classification in Table 1. (a) The confidence of the objects in group (a); (b) The confidence of the objects in group (b); (c) The confidence of the objects in group (c); (d) The confidence of the objects in group (d).



(a) (b)

FIGURE 9. The training loss of the YOLOv5 algorithm. (a) Loss comparison between the YOLOv5 and YOLOv4 algorithms; (b) Training loss for each iteration of the YOLOv5 algorithm. (YOLOv5 and YOLOv4 algorithms are marked with the blue line and Orange line, respectively.)

epoch 30. The cigarette butts eventually attain a steady condition after epoch 15, when their confidence is at its lowest. In Fig. 8 (b), while the book swings significantly and achieves the lowest confidence at epoch 40, the zip-top can and newspaper tend to be steady. The schoolbag remains stable until about epoch 16, at which point it begins to climb. The performance of the pill is at its poorest, as can be seen in Fig. 8 (c), reaching the lowest confidence at epoch 15 and below the confidence of 0.6. The performance of the other three things remained constant. According to Fig. 8 (d), fish bones and apple cores had the highest steady confidence levels. The confidence level for watermelon rind varies, falls at epochs 15 and 25, and reaches its lowest level at epoch 30. At epoch 40, the confidence of the eggshell is at its lowest point.

The training loss of the YOLOv5 algorithm is shown in Fig. 9. The loss comparison between the YOLOv5 and YOLOv4 algorithms under the same circumstances is shown in Fig. 9 (a). The YOLOv5 method has a smaller total loss than the YOLOv4 algorithm. At epoch 1, the YOLOv5 algorithm has a maximum training loss of 0.9 and a minimum loss of 0.014. At epoch 2 and epoch 43, respectively, the YOLOv4 algorithm hits its greatest training loss of 1.7371 and its minimum loss of 0.11737. The YOLOv5 algorithm's training loss and iteration count are related, as shown in Fig. 9 (b). As the number of iterations in each iteration increases, the training loss of the YOLOv5 algorithm reduces. It reaches its least loss at iteration 900, which corresponds to the smallest loss of the YOLOv5 algorithm at epoch 50 in Fig. 9 (a).

Classification loss and bounding box regression loss make up the loss function for tasks involving object recognition.

TABLE 2. The comparison between the YOLOv5 and YOLOv4 algorithms where trained in our recognition dataset corresponds to the data in Fig. 9.

Models	Minimum training loss	Epochs	Precision (%)	Model size (MB)	FPS
YOLOv5	0.014	50	96	27	120
YOLOv4	0.11737	43	94	244	50

TABLE 3. Results of ablation experiments. (Different attention modules are added to YOLOv5 to detect the performance of the models, where P2 represents the additional feature detection head).

Scheme	Models	Precision (%)	Recall (%)	mAP@0.5 (%)	Inference time (ms)
A	YOLOv5	76.0	60.1	64.5	10.5
B	YOLOv5+P2 [36]	81.8	62.5	67.5	10.5
C	YOLOv5+P2+SimAM [37]	90.0	88.1	88.7	10.5
D	YOLOv5+P2+CBAM [38]	93.3	90.3	94.7	3.6
E	YOLOv5+P2+PPO(ours)	96.0	89.2	92.3	1.2

The intersection of union (IoU), which is generated by computing the intersection ratio between the predicted boundary box and the actual boundary box, is the boundary box regression loss that is most frequently employed. Precision [34], recall, average precision (AP), and mean average precision (mAP) are some of the evaluation indices for model accuracy.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{14}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{15}$$

$$\mathrm{AP} = \frac{\Sigma \mathrm{P}}{Num(objects)} \tag{16}$$

$$\mathrm{mAP} = \frac{\Sigma \mathrm{AP}}{Num(class)} \tag{17}$$

where P stands for precision and R for recall, true positive (TP) is the proportion of correctly predicted positive samples, false positive (FP) is the proportion of correctly predicted negative samples, and false negative (FN) is the proportion of incorrectly predicted negative samples. Mean average precision (mAP) is for the mean values of all average precision (AP), and AP stands for the area under the precision-recall (PR) curve.

The accuracy of identifying each object in an image is measured with precision. In our object recognition grasping experiments, the YOLOv5 and YOLOv4 models were used to train the recognition datasets, and the comparison results obtained are shown in Table 2. Based on the block dataset developed in this paper, the YOLOv5 algorithm's minimal loss is 0.014 when the batch size is 16 and the epochs are 50. The training loss was reduced by 88.07% using the YOLOv5 algorithm as compared to the YOLOv4 algorithm. The mobile manipulator based on the YOLOv5 algorithm has a recognition precision of 96%, whereas the mobile manipulator based on the YOLOv4 method has a recognition precision of 94%. The accuracy of YOLOv5 has increased by 2.12% when compared to YOLOv4. YOLOv5 is merely 27 MB in size, compared to 244 MB for YOLOv4. The YOLOv5 detection speed, which is substantially faster than the YOLOv4 identification method, was 120 frames per second (FPS) [35]. The outcomes demonstrate the ability of the YOLOv5 vision algorithm to recognize and detect quickly and effectively with a small size.

C. ABLATION EXPERIMENT

To assess the effectiveness and advancement of the proposed algorithm in this paper, five groups of ablation experiments were conducted under the same validation set to evaluate the detection performance of different attention modules on the algorithm. Under the same experimental conditions, the precision (P), recall (R), mean average precision (mAP)@0.5, and inference time of each model were used to evaluate the impact of different modules on the object recognition algorithm of YOLOv5.

First, we studied the ablation experiment in which different lightweight feature extraction modules were added to the YOLOv5 framework, as shown in Table 3. We add a detection head with additional feature layers to the module, denoted by the P2 notation. Widely used lightweight attention modules are GhostConv, Simple Attention Module (SimAM), Convolutional Block Attention Module (CBAM), etc. The addition of the attention module can optimize the parameters of the model, mainly because the introduction of P2 changes the entire architecture of YOLOv5, making it better than other local modules. Therefore, robot recognition grasping requires more detection features and reduces the computational complexity of training in detection.

According to the data in Table 3, the results show that after adding P2 to the YOLOv5 network, compared with the initial YOLOv5 model, the precision improved by 5.8%, the recall improved by 2.4 percentage points, the mean average precision (mAP)@0.5 improved by 3 percentage points, and the inference time was 10.5 ms. Specifically, in the third group of experiments, after integrating the attention module Simple Attention Module (SimAM) into YOLOv5, the precision improved by 8.2%, and the recall

TABLE 4. The performance results of different YOLOv5 detection models on our recognition dataset.

Models	Precision (%)	Recall (%)	mAP@0.5 (%)	Model size (MB)	FLOPs (G)	Inference time (ms)
YOLOv5m	86.4	86.5	91.0	43.0	49.0	1.7
YOLOv5l	87.0	88.6	92.1	46.5	109.1	2.7
YOLOv5x	86.5	85.6	90.7	86.7	205.7	4.8
YOLOv5_ours	96.0	89.2	92.3	27.0	15.8	1.2

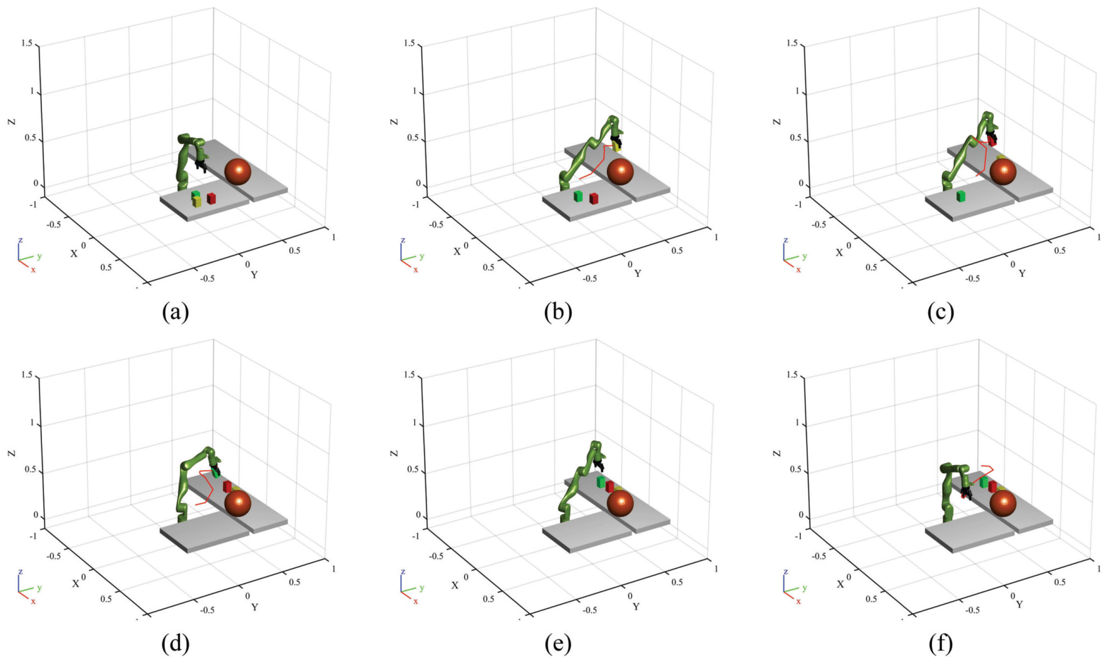
and mAP@0.5 significantly improved by 25.6% and 21.2%, respectively. It can be seen that the introduction of lightweight recognition modules has a great positive effect on the detection model. Specifically, in the fourth group of experiments, after integrating the attention module Convolutional Block Attention Module (CBAM) into YOLOv5, the precision is increased by 3.3 percentage points, and the recall rate and mean average precision (mAP)@0.5 are increased by 2.2 percentage points and 6 percentage points, respectively. The final results demonstrate that the method presented in this paper shows satisfactory performance, the precision of our algorithm is improved by 2.7%, and the inference time is shortened by 66.6% to 1.2 ms. Although the recall and mean average precision (mAP)@0.5 are reduced, the model recognition precision is greatly improved, indicating that our method can improve the object recognition ability and meet the requirements of real-time detection.

To further analyze the recognition performance of the models proposed in this paper, the YOLOv5m, YOLOv5l, YOLOv5x, and YOLOv5_ours models are used to detect our recognition dataset. The performance results of different YOLOv5 detection models are shown in Table 4. Precision, recall, mean average precision (mAP)@0.5, model size, floating point operations per second (FLOPs), and inference time are used as evaluation metrics. The results show that the precision, recall, and mAP@0.5 of our proposed method are 96%, 89.2%, and 92.3%, respectively. As can be seen from Table 4, the model of YOLOv5_ours in this paper has the highest precision, recall, and mAP@0.5 values. The precision values are 9.6%, 2.7%, and 1.3% higher than YOLOv5m, YOLOv5l, and YOLOv5x, respectively. The recall values are 2.7%, 0.6%, and 3.6% higher than YOLOv5m, YOLOv5l, and YOLOv5x, respectively. Mean average precision (mAP)@0.5 values are 1.3%, 0.2%, and 1.6% higher than YOLOv5m, YOLOv5l, and YOLOv5x, respectively. It shows that the YOLOv5_ours model has the best object detection effect among the four methods. The size of the proposed YOLOv5_ours model is only 27 MB. Compared with YOLOv5m, YOLOv5l, and YOLOv5x, the model is reduced by 37.2%, 41.9%, and 68.8%, respectively, achieving lightweight.

D. OBJECT GRASPING RESULTS

To train the grasping strategy, we use the Proximal Policy Optimization (PPO) reinforcement learning algorithm in the simulated environment. The trajectory results of object grasping are shown in Fig. 10. Fig. 10 (a) shows the preparation process for the mobile manipulator to start grasping. In its initial position, the mobile manipulator is prepared to snag the three color blocks on the plank. The orange ball in the illustration is a fictitious obstruction. To seize the block, the mobile robot arm must avoid the obstruction and place it on the left goal board. The mobile manipulator detects the object and transmits the block's coordinate information via communication with the Robot Operating System (ROS) server. Inverse kinematics is used to resolve the rotation angle of each joint. The mobile manipulator is propelled to grip when the joint angle has been determined. The mobile manipulator recognizes the object and sends out the information about the coordinates of the acquired square through Robot Operating System (ROS) server communication, and then solves the angle that each joint should rotate by inverse kinematics to obtain the joint angle and drive the mobile manipulator to grasp it. Fig. 10 (b) shows the mobile manipulator grasping the yellow block. After training to avoid obstacles and moving to the goal location, the mobile manipulator elevates the arm, releases the gripper to move to the yellow block position, clamps the yellow block, and then releases the gripper to reset. Fig. 10 (c) shows the mobile manipulator grasping the red block. Fig. 10 (d) shows the mobile manipulator grasping the green block. Fig. 10 (e) shows the end of the grasping task of the mobile manipulator, from which the three blocks can be seen placed at the target position. Fig. 10 (f) shows the reset of the gripper after the mobile manipulator has performed the grasping task. Among them, the gripper trajectory is shown in Fig. 11. When the mobile manipulator is reset, the gripper moves from the positive direction of the Y-axis to the negative position and from the negative direction of the X-axis to the positive position.

To further demonstrate the dependability of the Proximal Policy Optimization (PPO) grasping method, we contrasted the PPO algorithm with the Soft Actor-Critic (SAC) algorithm and the Trust Region Policy Optimization (TRPO) algorithm, as shown in Table 5. First, we train the algorithm under the same conditions with a batch size of 16. The data in Table 5 shows that the Proximal Policy Optimization (PPO) algorithm performs the best at batch 16. The PPO algorithm has the highest average reward and the lowest average step. The average reward of the Trust Region Policy Optimization (TRPO) algorithm is comparable to the PPO algorithm, and it has the second-highest performance. Although the Soft Actor-Critic (SAC) algorithm performs worse than the PPO and TRPO algorithms, it has the least



(a) (b) (c)

(d) (e) (f)

FIGURE 10. The trajectory results of object grasping via the PPO algorithm. (a) Start grasping; (b) Grasping the yellow block; (c) Grasping the red block; (d) Grasping the green block; (e) Finish grasping; (f) Reset manipulator's gripper. (Trajectories are marked with red line.)

TABLE 5. Compared the PPO algorithm with SAC and TRPO algorithms. (The average reward, average step and train time of the three algorithms are analyzed in batches 16 and 128, respectively).

Batch size	Algorithms	Average reward	Average step	Train time (h)
Batch = 16	PPO	-0.5414	1.0000	0.91
	SAC	-6.3294	41.3333	0.99
	TRPO	-0.5466	1.0000	0.93
Batch = 128	PPO	-0.3476	1.0000	0.55
	SAC	-5.2417	41.0667	0.61
	TRPO	-0.5888	1.0000	0.57

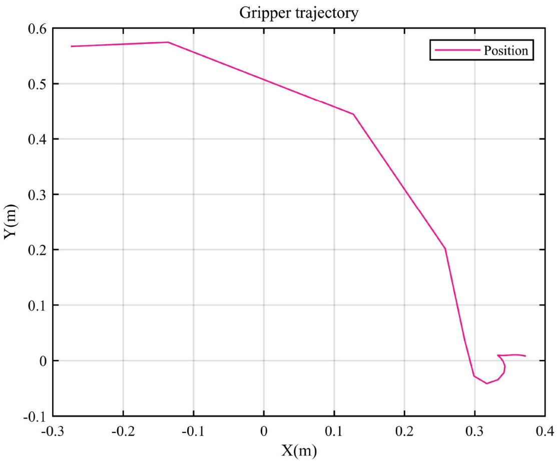


FIGURE 11. The position trajectory of the manipulator's gripper corresponds to the trajectory in Fig. 10 (f). (Position trajectory is marked with a pink line.)

average reward and the greatest average step. Compared with the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms, the average reward of the Proximal Policy Optimization (PPO) algorithm is improved by 91.4% and 0.95%, respectively. The PPO algorithm has the shortest training time compared with the SAC and TRPO algorithms, the training time is reduced by 8.08% and 2.15%, respectively.

Second, we train the algorithm's performance under identical conditions with a batch size of 128 to demonstrate the performance comparison. The Proximal Policy Optimization (PPO) algorithm performs the best at batch 128 as well. The PPO algorithm has the highest average reward and the smallest average step. The Trust Region Policy Optimization (TRPO) algorithm comes in second place in terms of performance, and it offers much higher average rewards than the PPO algorithm. Compared with the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms, the average reward of the Proximal Policy Optimization (PPO) algorithm is improved by 93.3% and 41%, respectively. The PPO algorithm has the shortest training time, which is reduced by 9.83% and

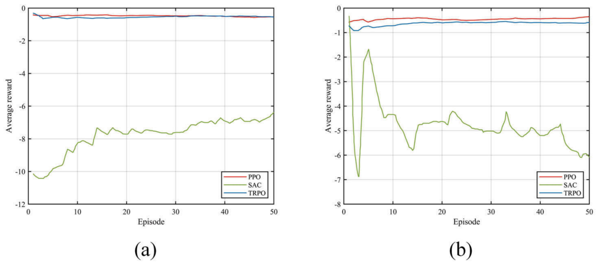


FIGURE 12. The average reward comparison of the PPO, SAC, and TRPO algorithms. (a) Average reward curve at batch 16; (b) Average reward curve at batch 128. (PPO, SAC, and TRPO algorithms are marked with red, green, and blue lines, respectively.)

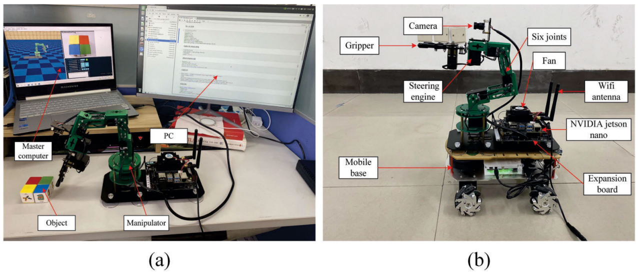


FIGURE 13. The experimental platform for object recognition and grasping. (a) Experimental platform setup; (b) The Physical model of the DOFBOT mobile manipulator. (The hardware devices of the mobile manipulator are marked with red arrows, respectively.)

3.50% compared with the SAC and TRPO algorithms, respectively.

Last but not least, Fig. 12 displays the average reward comparison of the PPO, SAC, and TRPO algorithms. The reward curve for batch 16 is displayed in Fig. 12 (a). The SAC algorithm has the lowest average reward, whereas the TRPO algorithm's average reward is comparable to that of the PPO algorithm. The reward curve for batch 128 is shown in Fig. 12 (b). The PPO algorithm provides the highest average reward and the best performance.

E. EXPERIMENTAL PLATFORM

A self-built Mecanum wheel vehicle serves as the mobile basis for the experiment's DOFBOT arm. Fig. 13 shows the experiment platform for object gripping. In Fig. 13 (a), the left side is a master computer, which is used to debug the real-time simulation of the manipulator. A personal computer (PC) on the right side is used to interact with the manipulator and debug the JupyterLab code. The manipulator is supported and powered by the mobile base. The gripper, steering engine, and camera on the manipulator are used to recognize and identify objects. The gripper is also used to pick up objects.

The Robot Operating System (ROS) operating system and Ubuntu 18.04 are installed on the DOFBOT arm. Based on Python, it can accomplish 472 Giga Floating Point Operations Per Second (GFLOPs) of computation [39]. The manipulator measures 272 × 135 × 473 mm in size. The gripper's positioning accuracy is just ±0.5 mm, and it can grasp objects weighing up to 500 g within a 30 cm radius.

TABLE 6. The grasping success rate of our approach. (The manipulator repeatedly identifies and grabs the object ten times, and places it at the target position, which means success).

Objects	Size (mm)	Success rate (%)
Yellow block	30 × 30 × 30	100
Blue block	30 × 30 × 30	100
Red block	30 × 30 × 30	100
Green block	30 × 30 × 30	90

We further detect the success rate of object grasping. The recognition area is filled with four color blocks, each of which faces the manipulator. The manipulator repeatedly spots the objects, grasps them ten times, and positions them in the target location. Table 6 presents the grasping success rate, and the four color blocks are all the same size. Green blocks have a 90% grasping success rate, while the grasping success rates for yellow, blue, and red blocks can reach 100%.

The gripping experiment scenes are shown in the real world in Fig. 14. Fig. 14 (a) shows the mobile manipulator starting the experiment, the mobile base of the manipulator starting to move, bypassing the obstacle to reach the end position, and the target objects are four color squares. Fig. 14 (b) shows the mobile manipulator reaching the grasping platform, lifting the arm, and the jaws being released. Fig. 14 (c) shows the mobile manipulator training to grasp the yellow block. Fig. 14 (d) shows the mobile manipulator training to grasp the blue block. Fig. 14 (e) and Fig. 14 (f) show the mobile manipulator training to grasp the red and green blocks, respectively.

V. DISCUSSION

This study aims to solve the problems of single application scenarios and the low accuracy of traditional grasping techniques. In this study, we demonstrate that our method has higher object recognition accuracy and mean average precision (mAP)@0.5 value, and also has a higher object grasping success rate. To illustrate the effect of object recognition and grasping, we combine the YOLOv5 algorithm with the Proximal Policy Optimization (PPO) reinforcement learning algorithm to provide an effective approach. Collectively, our data demonstrate that our method has the highest precision and mAP@0.5 values of 96% and 92.3%, respectively, by comparison of ablation experiments. Our method improves the average reward by 93.3% and 41% compared to the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms, respectively.

In this study, compared with the literature [9], we added the object grasping task to the target detection, while the precision and mean average precision (mAP)@0.5 values of our final results are higher than theirs. In this study,

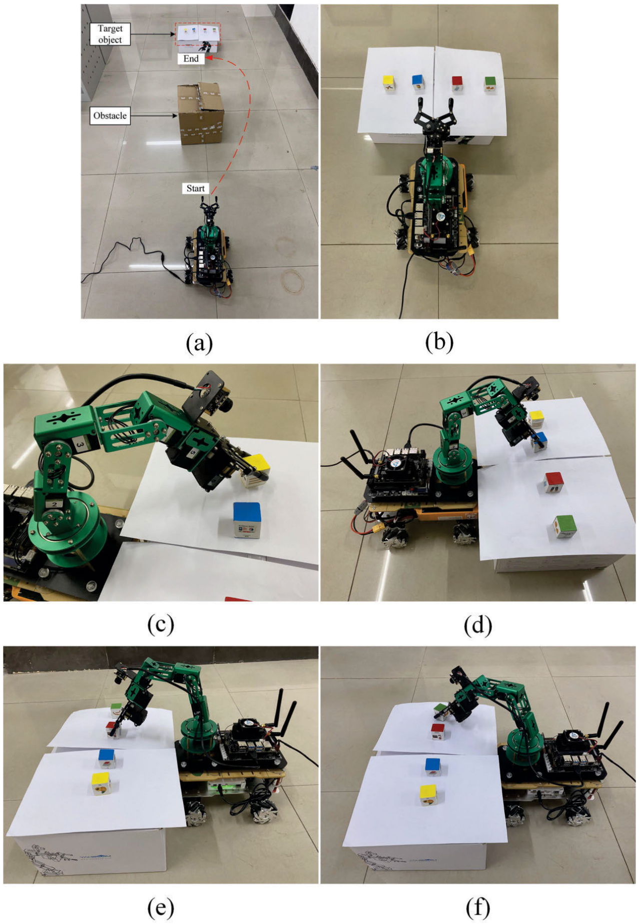


FIGURE 14. The grasping experiment scene in the real world. (a) Move the manipulator to the target point; (b) Mobile manipulator reaches the gripping platform; (c) Grasping the yellow block; (d) Grasping the blue block; (e) Grasping the red block; (f) Grasping the green block. (Four experimental color blocks are placed on the gripping platform.)

TABLE 7. The performance results for recognizing different objects. (The object categories correspond to the data in Table 1, and the evaluation indexes are precision, recall and mAP@0.5.)

Object category	Precision (%)	Recall (%)	mAP@0.5 (%)
Group (a) objects	93.1	75.6	88.9
Group (b) objects	90.2	80.8	90.1
Group (c) objects	89.1	92.9	96.8
Group (d) objects	88.4	87.4	93.5
All	96.0	89.2	92.3

compared with the literature [16], they combined the YOLOv5 method with the deep deterministic policy gradient (DDPG) reinforcement learning method to grasp the target, while we combined the YOLOv5 algorithm with the Proximal Policy Optimization (PPO) reinforcement learning algorithm, but we studied object recognition grasping, and the application range is wider than theirs. The performance results are more satisfactory. To make our method more effective, we recognize four groups of objects listed in Table 1, and the performance results are shown in Table 7. The proposed method shows effective performance in object recognition and grasping. The recognition precision and mean average precision (mAP)@0.5 values of our method are 96% and 92.3%, respectively. The precision and mAP@0.5 values of group (a) objects are 93.1% and 88.9%, respectively. The precision and mAP@0.5 values of group (b) objects are 90.2% and 90.1%, respectively. The recall and mAP@0.5 values of group (c) objects are 92.9% and 96.8%, respectively. The recall and mAP@0.5 values of group (d) objects are 87.4% and 93.5%, respectively. The computational complexity of our model is lower than that of traditional object grasping methods, and the precision is higher, enabling workers to accurately locate targets and accurately detect and grasp objects in industrial handling and agricultural picking fields. This study demonstrates an innovative combination of deep learning and reinforcement learning techniques to bridge the research gap in real-life applications of robots, especially in the context of the task of identifying and grasping objects by mobile manipulators. However, our study still has some limitations, such as blurred images due to the low pixel count of the camera of the physical manipulator. Therefore, the quality of the images needs to be improved, and a richer dataset is needed to enhance our detection performance.

VI. CONCLUSION

This paper proposes an object recognition grasping method using proximal policy optimization with YOLOv5, which combines vision recognition algorithms and deep reinforcement learning algorithms to achieve object recognition grasping. YOLOv5 is used for object recognition and obtains object location information, achieving higher accuracy and faster detection speed. The proximal policy optimization algorithm is used for object grasping to obtain the grasping strategy. The experimental results show that the proposed method has faster object recognition speed and higher detection accuracy. The maximum confidence of object recognition is 96%, and the minimum loss of the YOLOv5 algorithm is 0.014 under batch 16 and epoch 50. The YOLOv5 algorithm reduces the training loss by 88.07% compared to the YOLOv4 algorithm. The recognition precision of mobile manipulators based on YOLOv5 is 96%, while the recognition precision of YOLOv4 is 94%. Compared with YOLOv4, the precision of YOLOv5 is improved by 2.12%. By comparison of ablation experiments, our method has the highest precision and mean average precision (mAP)@0.5 value of 92.3%. Our proposed method outperforms the Soft Actor-Critic (SAC) and Trust Region Policy Optimization (TRPO) algorithms in object grasping. Compared with the SAC and TRPO algorithms, the average reward of the Proximal Policy Optimization (PPO) algorithm improves by 91.4% and 0.95% at batch 16, and by 93.3% and 41% at batch 128, respectively. In real physical experiments, we verify that the grasping success rate under our proposed approach reaches 100%.

To further verify the practical effectiveness of the proposed method in this paper, future research will replace the

PPO algorithm with other different reinforcement learning algorithms to improve grasping capability. The strategy network will also be trained using other techniques.

REFERENCES

[1] Y. Wang, L. Wang, and Y. Zhao, "Research on door opening operation of mobile robotic arm based on reinforcement learning," *Appl. Sci.*, vol. 12, no. 10, p. 5204, May 2022.

[2] A. Roennau, J. Mangler, P. Keller, M. Grosse Besselmann, N. Huegel, and R. Dillmann, "Grasping and retrieving unknown hazardous objects with a mobile manipulator," *Automatisierungstechnik*, vol. 70, no. 10, pp. 838–849, Oct. 2022.

[3] J.-R. Horng, S.-Y. Yang, and M.-S. Wang, "Self-correction for eye-in-hand robotic grasping using action learning," *IEEE Access*, vol. 9, pp. 156422–156436, 2021.

[4] S. Pan and T. Ahamed, "Pear recognition in an orchard from 3D stereo camera datasets to develop a fruit picking mechanism using mask R-CNN," *Sensors*, vol. 22, no. 11, p. 4187, May 2022.

[5] Q. Song, S. Li, Q. Bai, J. Yang, X. Zhang, Z. Li, and Z. Duan, "Object detection method for grasping robot based on improved YOLOv5," *Micromachines*, vol. 12, no. 11, p. 1273, Oct. 2021.

[6] Q. Bai, S. Li, J. Yang, Q. Song, Z. Li, and X. Zhang, "Object detection recognition and robot grasping based on machine learning: A survey," *IEEE Access*, vol. 8, pp. 181855–181879, 2020.

[7] C. Sun, Y. Ai, S. Wang, and W. Zhang, "Mask-guided SSD for small-object detection," *Int. J. Speech Technol.*, vol. 51, no. 6, pp. 3311–3322, Jun. 2021.

[8] Y. Wang and G. Fu, "A novel object recognition algorithm based on improved YOLOv5 model for patient care robots," *Int. J. Humanoid Robot.*, vol. 19, no. 2, Jul. 2022, Art. no. 2250010.

[9] S. Rani, D. Ghai, and S. Kumar, "Object detection and recognition using contour based edge detection and fast R-CNN," *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 42183–42207, Jul. 2022.

[10] X. Wang, X. Jiang, J. Zhao, S. Wang, and Y.-H. Liu, "Grasping objects mixed with towels," *IEEE Access*, vol. 8, pp. 129338–129346, 2020.

[11] P. Song, J. A. C. Ramón, and Y. Mezouar, "Dynamic evaluation of deformable object grasping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4392–4399, Apr. 2022.

[12] P. Chen and W. Lu, "Deep reinforcement learning based moving object grasping," *Inf. Sci.*, vol. 565, pp. 62–76, Jul. 2021.

[13] G. Zuo, J. Tong, Z. Wang, and D. Gong, "A graph-based deep reinforcement learning approach to grasping fully occluded objects," *Cognit. Comput.*, vol. 15, no. 1, pp. 36–49, Aug. 2022.

[14] H. Zhang, J. Tan, C. Zhao, Z. Liang, L. Liu, H. Zhong, and S. Fan, "A fast detection and grasping method for mobile manipulator based on improved faster R-CNN," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 47, no. 2, pp. 167–175, Jan. 2020.

[15] Z. Chang, L. Hao, H. Tan, and W. Li, "Design of mobile garbage collection robot based on visual recognition," in *Proc. IEEE 3rd Int. Conf. Autom., Electron. Electr. Eng. (AUTEEE)*, Shenyang, China, Nov. 2020, pp. 448–451.

[16] H. Sekkat, S. Tigani, R. Saadane, and A. Chehri, "Vision-based robotic arm control algorithm using deep reinforcement learning for autonomous objects grasping," *Appl. Sci.*, vol. 11, no. 17, p. 7917, Aug. 2021.

[17] G. Cao, X. Zhao, C. Ye, S. Yu, B. Li, and C. Jiang, "Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot," *J. Mech. Sci. Technol.*, vol. 36, no. 4, pp. 2019–2029, Apr. 2022.

[18] A. Gfrerrer, "Geometry and kinematics of the mecanum wheel," *Comput. Aided Geometric Design*, vol. 25, no. 9, pp. 784–791, Dec. 2008.

[19] Y. Han, Z. Chu, and K. Zhao, "Target positioning method in binocular vision manipulator control based on improved Canny operator," *Multimedia Tools Appl.*, vol. 79, nos. 13–14, pp. 9599–9614, Apr. 2020.

[20] Z.-D. Zhang, M.-L. Tan, Z.-C. Lan, H.-C. Liu, L. Pei, and W.-X. Yu, "CDNet: A real-time and robust crosswalk detection network on Jetson nano based on YOLOv5," *Neural Comput. Appl.*, vol. 34, no. 13, pp. 1–12, Feb. 2022.

[21] W. R. P. Novak, "Investigating evolutionary relationships through cluster analysis: A teaching science with big data workshop session," *Biochem. Mol. Biol. Educ.*, vol. 50, no. 5, pp. 440–445, Sep. 2022.

[22] Z. Wang, L. Wu, T. Li, and P. Shi, "A smoke detection model based on improved YOLOv5," *Mathematics*, vol. 10, no. 7, p. 1190, Apr. 2022.

[23] H. Qian and Y. Yu, "Derivative-free reinforcement learning: A review," *Frontiers Comput. Sci.*, vol. 15, no. 6, pp. 1–19, Sep. 2021.

[24] J.-D. Martn-Guerrero and L. Lamata, "Reinforcement learning and physics," *Appl. Sci.*, vol. 11, no. 8, p. 8589, Sep. 2021.

[25] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 28, 2020, doi: 10.1109/TNNLS.2022.3207346.

[26] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, M. A. D. S. L. Cuadros, and D. F. T. Gamarra, "Soft actor-critic for navigation of mobile robots," *J. Intell. Robot. Syst.*, vol. 102, no. 2, pp. 1–11, May 2021.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[28] Y. Gu, Y. Cheng, C. L. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 7, pp. 4600–4610, Jul. 2022.

[29] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *J. Intell. Manuf.*, vol. 33, no. 2, pp. 387–424, Feb. 2022.

[30] Z. Guo, W. Yang, M. Li, X. Yi, Z. Cai, and Y. Wang, "ALLIANCE-ROS: A software framework on ROS for fault-tolerant and cooperative mobile robots," *Chin. J. Electron.*, vol. 27, no. 3, pp. 467–475, May 2018.

[31] Z. Guan, C. Hou, S. Zhou, and Z. Guo, "Research on underwater target recognition technology based on neural network," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–12, Apr. 2022.

[32] T. Zhang, H.-M. Hu, and B. Li, "A naturalness preserved fast dehazing algorithm using HSV color space," *IEEE Access*, vol. 6, pp. 10644–10649, 2018.

[33] J. Kantner and I. G. Dobbins, "Partitioning the sources of recognition confidence: The role of individual differences," *Psychonomic Bull. Rev.*, vol. 26, no. 4, pp. 1317–1324, Aug. 2019.

[34] Y. Jin, H. Gao, X. Fan, H. Khan, and Y. Chen, "Defect identification of adhesive structure based on DCGAN and YOLOv5," *IEEE Access*, vol. 10, pp. 79913–79924, 2022.

[35] J. Yao, X. Fan, B. Li, and W. Qin, "Adverse weather target detection algorithm based on adaptive color levels and improved YOLOv5," *Sensors*, vol. 22, no. 21, p. 8577, Nov. 2022.

[36] H. Gong, T. Mu, Q. Li, H. Dai, C. Li, Z. He, W. Wang, F. Han, A. Tuniyazi, H. Li, X. Lang, Z. Li, and B. Wang, "Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images," *Remote Sens.*, vol. 14, no. 12, p. 2861, Jun. 2022.

[37] Q. Xiao, Q. Li, and L. Zhao, "Lightweight sea cucumber recognition network using improved YOLOv5," *IEEE Access*, vol. 11, pp. 44787–44797, 2023.

[38] S. Chen, Y. Liao, F. Lin, and B. Huang, "An improved lightweight YOLOv5 algorithm for detecting strawberry diseases," *IEEE Access*, vol. 11, pp. 54080–54092, 2023.

[39] A. Kumar, T. Kashiyama, H. Maeda, H. Omata, and Y. Sekimoto, "Real-time citywide reconstruction of traffic flow from moving cameras on lightweight edge devices," *ISPRS J. Photogramm. Remote Sens.*, vol. 192, pp. 115–129, Oct. 2022.

QINGCHUN ZHENG received the M.S. degree in agricultural mechanization from Huazhong Agricultural University, in 1996, and the Ph.D. degree in mechanical engineering from Tianjin University, China, in 2002.

He is currently a Professor with the School of Mechanical Engineering and the School of Computer Science and Engineering, Tianjin University of Technology. He is the author of more than 70 journal articles. His current research interests include computer vision, robot learning and intelligent control, and intelligent manufacturing. He is the Council Chairperson of the Tianjin Institute of Instrumentation.

ZHI PENG (Graduate Student Member, IEEE) received the B.S. degree in mechanical design manufacture and automation from Bengbu University, China, in 2021. He is currently pursuing the master's degree in mechanical engineering with the School of Mechanical Engineering, Tianjin University of Technology.

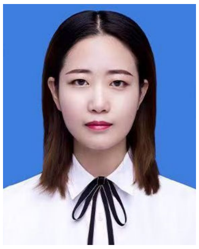
His current research interests include reinforcement learning, trajectory planning, and robot grasping.

PEIHAO ZHU received the M.S. degree in mechanical engineering from the Tianjin University of Technology, in 2007, and the Ph.D. degree in mechanical engineering from Tianjin University, China, in 2010.

He is currently an Associate Professor with the School of Mechanical Engineering, Tianjin University of Technology. His current research interests include robotics, artificial intelligence, and grinding automation.

YANGYANG ZHAO (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in automation and electrical engineering from the Shenyang Institute of Engineering, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in computer science and technology with the Tianjin University of Technology.

His current research interests include computer vision, multimodal fusion, and robot learning.



RAN ZHAI received the B.S. degree in mechanical process technology from Jinzhong University, China, in 2020. She is currently pursuing the master's degree in mechanical engineering with the School of Mechanical Engineering, Tianjin University of Technology.

Her current research interests include robot technology, vibration suppression, and mechanical technology.

WENPENG MA received the B.S. degree in mechanical engineering from Huazhong Agricultural University, in 2009, and the Ph.D. degree in mechanical engineering from Tianjin University, China, in 2015.

He is currently a Lecturer with the School of Mechanical Engineering, Tianjin University of Technology. His current research interests include robot control and vibration analysis.