An Efficient and Accurate DDPG-Based Recurrent Attention Model for Object Localization

FENGKAI KE

Hubei Key Laboratory of Modern Manufacturing Quality Engineering, School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China
e-mail: kfkhbut@163.com

This work was supported in part by the Scientific and Technological Research of Education Department of Hubei Province under Grant Q20181408, in part by the Scientific Research Foundation of Science and Technology Department of Hubei Province under Grant 2018CFB276, in part by the Doctor Launching Fund of Hubei University of Technology under Grant BSQD20160004, and in part by the Hubei Chenguang Talented Youth Development Foundation (HBCG) under Grant 2017109.

Abstract Using image processing algorithm to localize objects, which lack specific patterns and local features, has always been the research focus in industrial production. Compared with the traditional image processing algorithm, RAM (Recurrent Attention Model) in deep learning not only shows advantages in positioning accuracy and stability, but also has good adaptability in situations such as occlusion. However, RAM contains policy gradient (PG) algorithm, which is unstable in training process and has low convergence efficiency. To overcome this shortcoming, in order to improve the learning efficiency and stability of RAM, this paper proposes DDPG-based RAM. In addition, current random sampling algorithm in DDPG (Deep Deterministic Policy Gradient) does not make full use of the information contained in samples. Some samples are repeatedly learned, which slows down the convergence rate of the neural network model, and even causes the model to converge to the local optimal solution. To solve the above problems, a prioritized experience replay algorithm based on Gaussian sampling method is proposed. By constructing the localization and grasping simulation environment in V-rep, it is shown that compared with the traditional image algorithm, the proposed model algorithm in this paper has a greater improvement in localization accuracy, stability and model convergence speed.

Index Terms Experience replay, Gaussian sampling, image processing, RAM.

I. INTRODUCTION

With the development of automation, machine vision has gradually become an indispensable part of industrial production. In this regard, the use of vision-guided mobile robot or robotic arm for positioning has become one of the hot research areas. In industrial production, image-based robot arm positioning and grasping has become one of the measurement standards of automation level. Positioning and grasping is a very common problem in industrial production, but most of the existing visual technology is aimed at the recognition of objects with rich surface texture features.

Most of the industrial parts are smooth, and lack of specific texture or local features. In addition to the smooth surface and lack of texture features and local shape feature of most parts in industrial production, the problem of occlusion caused by randomly placed parts is also one of difficult problems to be considered. In view of these problems, this paper proposes to use the method of deep reinforcement learning, which only gives the pre-defined reward function and original image, to make the computer learn and recognize the positions of randomly placed parts and complete the subsequent grasping task.

In addition to the comparison with the traditional image processing algorithm for localization, this paper replaces the policy gradient (PG) algorithm in recurrent attention model (RAM) for deep deterministic policy gradient (DDPG) algorithm, and improves the experience replay method of DDPG by priority sampling.

The experimental results show that the proposed RAM model has better positioning precision and stability compared with the traditional localization algorithm. And the priority sampling algorithm further improves the learning efficiency of RAM.
II. RELATED WORK
A. RAM IN DEEP LEARNING
In industrial production, the parts have various shapes and features. The traditional image processing algorithm is not only complex in algorithm design, but also needs to be designed according to the characteristics of each object, which is tedious and time-consuming. The emergence of deep learning algorithm has greatly simplified this process. As a research direction of machine vision, deep learning has gradually entered the attention of researchers in various fields in recent years. To be honest, deep learning is not essentially a new research direction, it is partly based on the neural network. Traditional neural network can be divided into input layer, hidden layer and output layer [1]–[4]. Therefore, the traditional neural network has the disadvantages of too many parameters and poor convergence. The proposal of convolution neural network and recurrent neural network improves the traditional neural network in the number of layers, so as to extract more abstract and deeper features of the object, and improve the robustness and universality of the model for many applications. In addition, with the improvement of computer computing performance and the graphics processors, deep model learning becomes possible for common researchers. Now it has been gradually applied in image classification [5]–[7], expression recognition [8]–[10], speech recognition [11]–[14], etc.

Attention model is one of the research hot spots in deep learning model. The idea of creating attention model comes mainly from the human visual mechanism. When collecting images, human vision did not carefully observe all areas, but made quick scanning on the global image and then focused on the specific areas. So this model can output its own region of interest at each step. There are many kinds of applications of attention model, such as machine translation [15], image captioning [16], text summarization [17] and so on. There are two kinds of attention models, one is soft attention model and the other is hard attention model. Soft attention model is the weight ratio of the whole original input, so as to determine the focus of the model and weaken the area outside the focus [18]. In the hard attention model, the focus position is determined by reinforcement learning, and the out-of-focus area is ignored [19].

There have been many improvements to the attention model, such as self-attention model [20], residual attention model [21], and some module optimization in the attention model, such as GRU [22] and convolution kernel optimization [23]. Although there are some improvements in performance, they are only limited to the fine tuning of the model, and there is no in-depth analysis from the overall structure and operation mode of the model.

For the application studied in this paper, the main purpose of industrial production is to quickly and accurately process image data in real time. Compared with the soft attention model, the hard attention model only intercepts parts of the input image. It can quickly analyze and output it. Therefore, according to the processing mechanism of the hard attention model, this paper optimizes its learning process and proposes a priority sampling algorithm.

B. REINFORCEMENT LEARNING IN RAM
Reinforcement learning is considered as one of the main development paths of artificial intelligence [24], [25]. In 2015, deep reinforcement learning was first successfully applied in Atari2600 game, from Deep Q Network (DQN) proposed by Mnih [26]. The performance of the algorithm proposed in this paper exceeds that of ordinary human players in 40 games. Among them, it surpasses professional players in 34 games, and surpasses the most advanced intelligent algorithm at that time in at least 28 Games. The success of this model lies in two aspects: (1) the target network and the evaluation network; (2) experience pool. In the process of training, the method of sampling samples from the experience pool is random sampling. The disadvantage is that the information contained in the samples is not fully utilized, which reduces the efficiency of model training.

The agent of reinforcement learning accomplish a certain task through a large number of training. Reinforcement learning algorithms can be divided into model-based learning and model-free learning. The model-based learning algorithms are based on the transition probability model, which is to calculate the probability between the current state and the next state. However, the model-based learning algorithms are usually applied in simulation environment, which has the low dimension state space and low dimension action space, such as chessboard game and doby slot machines (multi-armed bandits). The model-free learning algorithms don’t use the transition probability model, but only use the samples in the process of task execution. The learning methods of samples are divided into online learning and offline learning. The online policy learning algorithms use the samples from the current policy for training. They does not need to store too many samples, thus they can avoid consuming too much computer memory. But this kind of algorithm is affected by the correlation between samples. The samples generated between two adjacent policies will produce different reward functions due to the influence of noise, which will lead to oscillation of the model learning process and even the failure to converge to the optimal solution. Therefore, off-line policy learning based on non-transition probability model is often adopted. Mnih proposed experience replay in offline policy learning. The off-line policy learning uses the experience pool to store the samples executed by the previous policy and make the agent learn from these samples repeatedly. It can not only reduce the correlation between samples, but also improve the model convergence probability. Both deep Q-network and Actor-Critic network consist of the experience replay algorithm. However, the experience replay algorithm uses random sampling, which does not fully utilize the information of samples, such as TD error, etc. Some samples that have been learned by models are still repeatedly selected for learning. The reverse gradient that these samples can produce is close to zero, which results in almost no updating of model
parameters. Therefore, in response to this problem, many researchers have proposed how to use the information of the samples to accelerate the neural network learning process. Zhai Jianwei et al. proposed to set the sampling priority based on positive and negative reward value, and use the number of sampling times, so as to take into account the sampling probability of various samples and ensure the diversity of samples in the learning process [27]. However, the disadvantage is that the reward value in the current time does not completely represent the cumulative reward or long-term benefit of the current action, but only indicates that there is a certain reward value for executing the action in this state. Moreover, in many tasks the agent can only get sparse reward, and even get the reward at the end of task. Therefore, this algorithm is not universal. Lin Ming et al. proposed an experience replay algorithm based on fixed length [28]. The main idea is to transform the traditional single-step length sample learning into fixed-length sample training. In his paper, it is proved that increasing training samples of fixed length can improve training efficiency in non-Markov environment. However, this kind of model requires a long period of sample storage, which is not efficient. It is shown that there is no significant positive effect in Markov environment, and sometimes even reduces the learning efficiency. Zhu Fei et al. proposed a deep Q network based on experience sampling of the maximum confidence upper bound. This algorithm is mainly based on the sampling idea of the maximum confidence upper bound, and uses the reward value for setting the priority. However, this algorithm still has the problem of unclear probability distribution of positive and negative reward values, and cannot fully guarantee the diversity of samples in the sampling process [29].

RAM integrates the strong nonlinear fitting ability of neural network and the advantages of model-free learning in reinforcement learning. Reinforcement learning is an essential part of RAM. According to the above problem, this paper proposes a prioritized experience replay algorithm based on the Gauss process. This model requires a long period of sample storage, which is not efficient. It is shown that there is no significant positive effect in Markov environment, and sometimes even reduces the learning efficiency. Zhu Fei et al. proposed a deep Q network based on experience sampling of the maximum confidence upper bound. This algorithm is mainly based on the sampling idea of the maximum confidence upper bound, and uses the reward value for setting the priority. However, this algorithm still has the problem of unclear probability distribution of positive and negative reward values, and cannot fully guarantee the diversity of samples in the sampling process [29].

As mentioned before, most of the existing improvements to the attention model are improving the structure other than PG method in Fig. 1, which cannot fundamentally improve the learning efficiency and stability of RAM. PG is one of the most basic strategy models of reinforcement learning. It does not have to occupy memory. However, its disadvantages are unstable learning process and poor convergence. This is one of the reasons for the slow learning efficiency and even failures of RAM. To solve this problem, the paper replace the original PG method with a more stable and efficient DDPG method, as shown in Fig. 2.

III. RAM BASED ON DDPG AND PRIORITIZED EXPERIENCE REPLAY ALGORITHM

A. RAM BASED ON DDPG

In order to avoid the design of feature extraction algorithm in traditional algorithm, this paper proposes to use RAM for object recognition to guide the robot arm to complete the corresponding grasping work. Compared with CNN, RAM can better fit the real-time requirements of industrial production by intercepting partial input to complete the target. The RAM used in this paper is a hard attention model, which is mainly composed of two parts. One part is the recurrent neural network, which extracts features and locates the final object by intercepting part of the input image, and the other part is the reinforcement learning network (e.g. PG in Fig. 1), which is used to output the location of the focus of the whole model at the next moment, as shown in Fig. 1.

B. PRIORITIZED EXPERIENCE REPLAY ALGORITHM

In addition to the improvement of RAM on the learning strategy model, this paper also makes an in-depth improvement on the learning method of DDPG itself, and proposes the prioritized experience replay sampling algorithm. In order to better express the algorithm proposed in this paper, the following are the notations and definitions involved in the algorithm.

Markov Decision Process (MDP) is usually used to simplify the modeling of reinforcement learning [30]. Markov
The decision process can usually be represented by a quaternion array, that is, \((s, a, p, r)\), where \(s\) is the state, \(a\) is the action performed in the current state, \(p\) is the probability of moving to the next state \(s'\) after taking the action, and \(r\) is the real-time reward value obtained when reaching the next state.

There are two learning methods that do not require the state transition probability model. One is the Monte Carlo (MC) method. The other method is Temporal Difference (TD). They do not need the transition probability model, but only need the “experience” in the environment, that is, the state, action and reward value.

The state value function \(V(s)\) is the average of the cumulative reward value after experiencing status \(s\), which is defined as follows

\[
V(s) = \text{average}(\text{return}(s))
\]

where \(\text{return}(\cdot)\) is the cumulative reward value after experiencing state \(s\). Similarly, the \(Q(s, a)\) value function is defined as

\[
Q(s, a) = \text{average}(\text{return}(Q(s, a)))
\]

where \(\text{return}(\cdot)\) is the cumulative reward value after experiencing state \(s\) and action \(a\).

MC method can predict the \(V\) value and \(Q\) value in two ways. One is to save the average of the cumulative rewards experiencing state \(s\) for the first time; The other is to save the average of cumulative reward experiencing state \(s\) for each time. According to the definition of MC method, \(V\) values and \(Q\) values need to be determined after each end of the task, which is not conducive to real-time update. TD method can converge to the real value more quickly in time. Unlike the MC method, TD method updates the \(V\) values and \(Q\) values incrementally, which are defined as follows

\[
V(s_t) \leftarrow V(s_t) + \alpha (r_{t+1} + \gamma V(s_{t+1}) - V(s_t))
\]

where \(\alpha\) is the incremental update coefficient of the state value function, indicating the trust degree of the agent to the new sample, \(\gamma\) is the attenuation coefficient of the future reward value, and represents the importance to the current reward value. Similarly, the \(Q\) value function of TD method can be defined as

\[
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha (r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))
\]

TD error is defined as the difference between the right and left sides of (3) and (4). TD error can evaluate the learning degree of an agent in the simulation environment. A small TD error can indicate that the agent has completed the learning under the current state and action.

According to the idea proposed by Schaul et al. [31], the TD error value is used for setting priority to sort the samples. However, if only based on the TD error values saved in the previous policy model, these TD error values can only indicate the learning status of the samples under the previous policy model. It can not represents the relationship between the previous and the current policy model. Therefore, in this paper, we calculate the TD error values under the previous policy model and the current policy model, denoted as \(TD_{error1}\) and \(TD_{error2}\) respectively.

When \(TD_{error2}\) is small, it indicates that the corresponding \(V\) value or \(Q\) value at current state and action is close to convergence, so there is no need to learn too much. Otherwise, it indicates that the sample needs to be further learned. In addition, the difference \(TD_{diff}\) between \(TD_{error1}\) and \(TD_{error2}\) also reflects the difference between the previous and current policy models, so it also needs to be taken into consideration. In order to avoid the same sample being sampled for many times, \(\beta^c\) the attenuation coefficient of the number of samples is added, namely, \(c\) is the number of samples, and \(\beta\) can be set to 0.95~0.99, so as to decrease the sampling priority of samples. Finally, the sample with high reward value indicates that it is close to the optimal policy and has the necessity to be selected first. Therefore, the specific definition of priority is as follows

\[
\text{rank} = f \left( TD_{error2} \right) 
\times TD_{diff} \times \beta^c \times r_t
\]

where \(r_t\) is the reward value of the sample, \(f(\cdot)\) can be defined as functions such as sigmoid to prevent the \(TD_{error2}\) value from affecting the priority too much, \(TD_{diff}\) is defined as the absolute value of the TD error difference between the old and new policy models, as shown below

\[
TD_{diff} = |TD_{error1} - TD_{error2}|
\]

In this paper, the follow-up experimental data show that using only the samples selected from (5) will affect the diversity of samples. Therefore, we propose to use Gaussian sampling method in each batch of training samples to ensure the diversity of samples in the learning process.

The prioritized experience replay algorithm based on Gaussian sampling method is shown in Table 1. The proposed method is based on DDPG algorithm. The DDPG algorithm has two main parts, Actor network and Critic network. Both of them have two network, evaluation network and target network. The evaluation network and target network are used for reducing the correlation between samples to ensure the stable convergence of the model. In the table, the memory_size is the number of samples to store in experience pool. \(M\) is the number of total episode, \(T\) is the number of total steps in each episode. \(N(\cdot)\) is the Gaussian probability model for sampling. \(f\) and \(K\) are used to control the update frequency of the target network \((\theta^\mu\) and \(\theta^\Omega\)) from the evaluation network \((\theta^\mu\) and \(\theta^\Omega\)).

### IV. EXPERIMENTS AND DATA ANALYSIS

In order to verify the feasibility of the algorithm, the Dobot robot positioning task is used under the V-rep simulation environment.

As shown in Fig. 3, the top left corner shows a panoramic view of the entire simulation environment, while the other three images are observed from different angles. Dobot is placed in the middle of the environment, a camera with
F. Ke: Efficient and Accurate DDPG-Based Recurrent Attention Model for Object Localization

TABLE 1. Prioritized experience replay algorithm based on Gaussian sampling method.

| Step | Content |
|------|---------|
| 1    | Constructing Actor evaluation network and Critic evaluation network, and randomly initializes their weights \( \mu(s|\theta^\mu) \) and \( Q(s,a|\theta^Q) \) respectively. Assign the evaluation network weight parameters to Actor target network and Critic target network, that is, \( \theta^\mu \leftarrow \theta^\mu \) and \( \theta^Q \leftarrow \theta^Q \). |
| 2    | Create the experience pool and set its size to \( \text{memory size} = (2 \times \text{state dim} + \text{action dim} + 3) \). |
| 3    | In order to improve the diversity of samples, the actions are chosen by Gaussian distribution sampling, \( a_t = \mathcal{N}(\mu(s_t|\theta^\mu), \text{var}) \). |

Enter into the training loop.

\( \text{For episode} = 1: M \)

1) Enter into initial state \( s_1 \).

2) For \( step = 1: T \)

   ① Input the current state \( s_t \) into the evaluation network \( \mu(s_t|\theta^\mu) \).

   ② Output the current action \( a_t = \mathcal{N}(\mu(s_t|\theta^\mu), \text{var}) \).

   ③ Execute action \( a_t \), then get reward \( r_t \) and next state \( s_{t+1} \).

   ④ Calculate the \( TD_{\text{error}} \) using the \( (s_t, a_t, r_t, s_{t+1}) \).

   ⑤ Save the sample in the experience pool and set its sample time \( c = 1 \).

1) Sample \( 3 \times \text{batch} \) samples, and calculate the \( TD_{\text{error}} \) for current policy.

2) Calculate the rank of priority using equation (5).

3) Sort the sample using the rank of priority.

4) Sample \( 1 \times \text{batch} \) in the middle of the \( 3 \times \text{batch} \) by Gaussian sampling.

⑥ When the experience pool is full, then start training.

⑦ Using the samples in the previous step to train \( \mu(s|\theta^\mu) \) and \( Q(s,a|\theta^Q) \).

⑧ Update the \( TD_{\text{error}} \) and the sample time \( c \) of the samples in \( \Theta \).

⑨ When updates the \( \mu(s|\theta^\mu) \) for \( J \) times, \( \theta^\mu \leftarrow \theta^\mu \).

⑩ When updates the \( Q(s,a|\theta^Q) \) for \( K \) times, \( \theta^Q \leftarrow \theta^Q \).

End for.

End for.

FIGURE 3. Dobot simulation environment.

A 60-degree angle of view and a resolution of \((640, 480)\) is placed directly above the robot, and a square object is randomly placed in the field of view of the camera. The task of the robot is to avoid the flower pot, grab the object and place them on the conveyor belt using only the original image from the camera. The workflow involves how to avoid part positioning problem, occlusion problem, flowerpot collision problem, kinematic inverse solution problem and so on.

In order to make the robot locate the object automatically and realize that it catches the object instead of randomly swinging to reach the target position, the reward function is designed as follows. When the end of the robot arm is not in the square area of the object, the reward function is defined as the negative value of the distance between the robot end and the center of the object. When the accurate positioning of 10 time steps is completed, the task is completed and the next episode restarts. If five successive episodes succeed, the experiment is stopped. An experiment contains 600 episodes. Each episode contains 200
steps. In the case of completing episode five times in a row, it shows that the robot has learned the task of positioning and completed the experiment. Record the episode after completing each experiment, and judge the learning efficiency of the proposed reinforcement learning algorithm by the mean number of episode. We have done 10 experiments for each algorithm to reduce randomness. Compared with the discrete action control problems, the continuous action control problems in this paper is more difficult to converge, which is suitable for verifying the feasibility of our proposed model and algorithm.

The hardware configuration of this paper is Intel i7-7700HQ quad-core processor, the main frequency is 2.7Ghz, 8G DDR4 2400MHz memory, the graphics card is NVIDIA GeForce 940MX. The system is Ubuntu 16.04, and Tensorflow 1.5 is used to build the neural network.

**A. TRADITIONAL OBJECT RECOGNITION AND LOCALIZATION**

In this paper, the traditional localization algorithm is also compared and analyzed. In Fig. 4, SIFT (Scale-Invariant Feature Transform) and FLANN (Fast Library for Approximate Nearest Neighbour) algorithm are used for feature extraction and feature matching algorithm to locate the object position.

The blue circles in the figure represent feature points, and the green lines connect the matched features. From the figure, we can see that although SIFT can detect many features, it is difficult to match them correctly. This is due to the smooth surface and the lack of obvious texture features of the measured object.

To overcome the above problems, the color adaptive threshold (CAT) algorithm and the minimum external rectangle (MER) algorithm are used to extract the center position of the part.

As shown in Fig. 5(a), is the detection effect of the adaptive threshold segmentation. The green area in the Fig. 5 is the result after positioning, and the result shows the accurate positioning result of the algorithm. However, when the occlusion, the algorithm can only locate the visible part, as shown in Fig. 5(b). It can be seen from Fig. 4 that SIFT + FLANN, a traditional feature detection and localization method, requires the detected object to have rich surface features. It is difficult to accurately locate the target in the simulation environment in this paper. Therefore, the experimental data of SIFT + FLANN are not listed in Table 2. In addition to Fig. 5, the relative positioning errors and standard deviation between different algorithms and models shown in Table 2. The mean value in Table 2 comes from the relative error value of positioning accuracy of the model (or algorithm) after training when the measured object is in different positions, and the variance is calculated by multiple mean error. The reduction of the average positioning error proves that the end effector of the robot arm is closer to the center of the measured object. On the premise of ensuring the reduction of mean error, the reduction of variance proves that the stability of the algorithm is improved. From the table, we can see that the traditional algorithm has poor adaptability in special cases.

**B. RAM AND PRIORITIZED EXPERIENCE REPLAY ALGORITHM**

Before outputting the coordinate value of the measured object, RAM needs to undergo four image processing, and each image comes from cutting part of the original image.
is completed. When completing the episode successfully the area of the measured object continuously, the episode the end effector of the manipulator can stay 10 steps in continuously in the cube area, the reward value is 10. When the peak value, the two algorithms fall back to a certain and PER algorithm proposed in this paper. After reaching algorithm shows certain volatility, especially for the DDPG seen from the figure that at the beginning of learning, each Here we only compare their convergence effect. It can be accepted by people. Therefore, the operation time of the a lot of time and training samples in advance, which can be seen from the figure, the proposed algorithm can converge be accepted by people. Therefore, the operation time of the a lot of time and training samples in advance, which can be seen from the figure, the proposed algorithm can converge faster and obtain higher cumulative reward than the other two algorithms. As for the PG algorithm which exists in original RAM, due to the influence of noise and the instability of the PG algorithm, it is difficult to converge, which is why the cumulative reward from 0 to 600 per episode is almost flat, because every episode is likely to converge. The difference is that the DDPG and PER proposed in this paper both had very low cumulative rewards before 50 episodes, but they quickly climbed to very high cumulative rewards after that and finally backed to around 0. This is because most of the two algorithm in the beginning of the accumulation of experience is wrong, it is difficult to make the right judgment. When the correct experience appears, it will be repeatedly compared with the previous strategy, so as to quickly complete the training. In addition, the learning curve of PER (Center) is similar to that of PER (Gaussian). However, PER (Center) algorithm ignores the diversity of samples and only selects the middle part of samples, it is slightly inferior to PER (Gaussian) in convergence speed and cumulative reward. According to the grasping experiment results, 600 episodes are enough for the robot to learn grasping skills. Once the robot has learned the grasping skill, it will terminate the experiment ahead of time. This is why the reward value in the second half of the learning curve of each algorithm in Fig. 7 tends to zero. According to the experimental data, the average convergence rounds of PG, DDPG, PER(Center) and PER(Gaussian) are 382.4, 296.2, 275.6 and 237.7, respectively. Therefore, the experimental results show that prioritized experience replay algorithm proposed in this paper is better to converge to the optimal solution compared to the traditional PG and DDPG algorithm.

V. CONCLUSION

In this paper, a deep learning model RAM is used to locate and identify the objects, which is placed on the ground randomly. The surface of the object is smooth and lacks texture features. We compared with localization performance of the traditional algorithms SIFT+FLANN and CAT+MER with RAM. Deep learning model RAM has great advantages in object recognition, occlusion and so on. We also propose DDPG-based RAM to solve the problem of poor convergence of PG algorithm in the original RAM. In addition, for the experience replay algorithm involved in DDPG, we also propose a priority experience replay algorithm based on Gaussian sampling. Experimental results show that compared with PG algorithm, DDPG and PER algorithm improve the convergence speed and convergence effect of the original model. The PER algorithm has a great improvement in the learning priority of samples, which improves the training efficiency and avoids the model convergence to the local optimal solution.

The algorithm proposed in this paper has obtained good preliminary research results in the simulation environment of Dobot robot. Future research will be improved in the following aspects: how to apply the model to the real application,

![FIGURE 7. Different versions of RAM learning curve.](image)

As shown in Table 2, after model training, DDPG+ PER has better positioning accuracy and stability than other two algorithms. As shown in Fig. 7, the four algorithms have been tested for 10 times in the simulation environment, the envelope of each curve in the figure is the 95% confidence interval of the algorithm’s cumulative reward for each episode. PG, DDPG and PER represent the RAM using PG, DDPG and DDPG+PER, respectively. In order to verify the necessity of using Gaussian sampling algorithm in PER algorithm proposed in this paper, two different sampling methods are used, one is to select the second batch of 3 batches directly in the training process, which is named PER (Center); the other is the Gaussian sampling method (as shown in Table 1), which is marked as PER (Gaussian).

Most of the deep neural networks need to learn through a lot of time and training samples in advance, which can be accepted by people. Therefore, the operation time of the algorithm is not within the scope of our measurement index. Here we only compare their convergence effect. It can be seen from the figure that at the beginning of learning, each algorithm shows certain volatility, especially for the DDPG and PER algorithm proposed in this paper. After reaching the peak value, the two algorithms fall back to a certain extent. From the definition of the reward function, it can be seen that when the end of the robot arm is stably and continuously in the cube area, the reward value is 10. When the end effector of the manipulator can stay 10 steps in the area of the measured object continuously, the episode is completed. When completing the episode successfully 5 times in a row, the task will be terminated in advance. As can be seen from the figure, the proposed algorithm can converge faster and obtain higher cumulative reward than the other two algorithms. As for the PG algorithm which exists in original RAM, due to the influence of noise and the instability of the PG algorithm, it is difficult to converge, which is why the cumulative reward from 0 to 600 per episode is almost flat, because every episode is likely to converge. The difference is that the DDPG and PER proposed in this paper both had very low cumulative rewards before 50 episodes, but they quickly climbed to very high cumulative rewards after that and finally backed to around 0. This is because most of the two algorithm in the beginning of the accumulation of experience is wrong, it is difficult to make the right judgment. When the correct experience appears, it will be repeatedly compared with the previous strategy, so as to quickly complete the training. In addition, the learning curve of PER (Center) is similar to that of PER (Gaussian). However, PER (Center) algorithm ignores the diversity of samples and only selects the middle part of samples, it is slightly inferior to PER (Gaussian) in convergence speed and cumulative reward. According to the grasping experiment results, 600 episodes are enough for the robot to learn grasping skills. Once the robot has learned the grasping skill, it will terminate the experiment ahead of time. This is why the reward value in the second half of the learning curve of each algorithm in Fig. 7 tends to zero. According to the experimental data, the average convergence rounds of PG, DDPG, PER(Center) and PER(Gaussian) are 382.4, 296.2, 275.6 and 237.7, respectively. Therefore, the experimental results show that prioritized experience replay algorithm proposed in this paper is better to converge to the optimal solution compared to the traditional PG and DDPG algorithm.

V. CONCLUSION

In this paper, a deep learning model RAM is used to locate and identify the objects, which is placed on the ground randomly. The surface of the object is smooth and lacks texture features. We compared with localization performance of the traditional algorithms SIFT+FLANN and CAT+MER with RAM. Deep learning model RAM has great advantages in object recognition, occlusion and so on. We also propose DDPG-based RAM to solve the problem of poor convergence of PG algorithm in the original RAM. In addition, for the experience replay algorithm involved in DDPG, we also propose a priority experience replay algorithm based on Gaussian sampling. Experimental results show that compared with PG algorithm, DDPG and PER algorithm improve the convergence speed and convergence effect of the original model. The PER algorithm has a great improvement in the learning priority of samples, which improves the training efficiency and avoids the model convergence to the local optimal solution.

The algorithm proposed in this paper has obtained good preliminary research results in the simulation environment of Dobot robot. Future research will be improved in the following aspects: how to apply the model to the real application,
without damaging the corresponding robot equipment; how to add dimension evaluation mechanism for faster training and learning.

REFERENCES

[1] J. A. M. Kamarudin, A. Abdullah, and R. Sallehuddin, “A review of deep learning architectures and their application,” in Proc. Asian Simulation Conf., 2017, pp. 83–94.

[2] J. Fu, H. Zheng, and T. Mei, “Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 4476–4484.

[3] Y. Zhang and B. Shi, “Improving pooling method for regularization of convolutional networks based on the failure probability density,” Optik, vol. 145, pp. 258–265, Sep. 2017.

[4] B. Wu, S. Han, J. Xiao, X. Hu, and J. Fan, “Error compensation based on BP neural network for airborne laser ranging,” Optik, vol. 127, no. 8, pp. 4083–4088, Apr. 2016.

[5] K. Wu, X. Chen, and M. Ding, “Deep learning based classification of focal liver lesions with contrast-enhanced ultrasound,” Optik, vol. 125, no. 15, pp. 4057–4063, Aug. 2014.

[6] Z. Yang, X.-D. Liu, and F.-A. Zhao, “Scene classification of remote sensing image based on deep network and multi-scale features fusion,” Optik, vol. 171, pp. 287–293, Oct. 2018.

[7] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, “3-D deep learning approach for remote sensing image classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[8] G. Chen, Y. Shao, C. Tang, Z. Jin, and J. Zhang, “Deep transformation learning for face recognition in the unconstrained scene,” Mach. Vis. Appl., vol. 29, no. 10, pp. 513–523, 2018.

[9] K. Shailaja and B. Anuradha, “Effective face recognition using deep learning based linear discriminant classification,” in Proc. IEEE Int. Conf. Comput. Intel. Comput. Res. (ICICIC), Dec. 2016, pp. 15–17. [Online]. Available: https://ieeexplore.ieee.org/document/7919708

[10] A. Vinay et al., “Deep learning on binary patterns for face recognition,” Procedia Comput. Sci., vol. 132, pp. 76–83, Jun. 2018.

[11] J. Zhang and C. Zong, “Deep learning for natural language processing,” in Deep Learning: Fundamentals, Theory and Applications. Springer, 2019, pp. 111–138.

[12] H. Yang et al., “Deep learning and its applications to natural language processing,” in Deep Learning: Fundamentals, Theory and Applications. Springer, 2019, pp. 89–100.

[13] C. Wu, M. J. F. Gales, A. Ragni, P. Karanasou, and K. C. Sim, “Improving interpretability and regularization in deep learning,” IEEE/ACM Trans. Audio, Speech, Language Process., vol. 26, no. 2, pp. 256–265, Feb. 2018.

[14] S. Hoo da and L. Kosseim, “Argument labeling of explicit discourse relations using LSTM neural networks,” in Proc. RANLP-Recent Adv. Natural Lang. Process., 2017, pp. 309–315. [Online]. Available: https://arxiv.org/abs/1708.03425v1

[15] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in Proc. Empirical Methods Natural Lang. Process., 2015, pp. 1412–1421, doi: 10.18653/v1/D15-1166.

[16] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, Jun. 2016, pp. 4651–4659.

[17] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proc. Conf. Empirical Methods Natural Lang. Process., Sep. 2015, pp. 379–389.

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in Proc. ICML, 2015, pp. 2048–2057.

[19] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst., vol. 2, 2014, pp. 2204–2212.

[20] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “DiSAN: Directional self-attention network for RNN/CNN-free language understanding,” 2017, arXiv:1709.04696. [Online]. Available: http://arxiv.org/abs/1709.04696

[21] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, “Residual attention network for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6450–6458.

[22] B. Zhang, D. Xiong, and J. Su, “A GRU-gated attention model for neural machine translation,” 2017, arXiv:1704.08430. [Online]. Available: http://arxiv.org/abs/1704.08430

[23] W. Ji-Li, P. Dunlu, C. Zhang, and L. Cong, “AM-CNN: A convolution neural network architecture for text classification based on attention mechanism,” J. Chin. Mini-Micro Comput. Syst., vol. 40, no. 4, pp. 710–714, 2019.

[24] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, “ActiVis: Visual exploration of industry-scale deep neural network models,” IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 1, pp. 88–97, Jan. 2018.

[25] J. Berg and K. Nystro¨m, “A unified deep artificial neural network approach to partial differential equations in complex geometries,” Neurocomputing, vol. 317, pp. 28–41, Nov. 2018. [Online]. Available: https://arxiv.org/abs/1711.06464

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, M. G. Bellemare, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[27] J. W. Zhao, “The research of algorithm and architectures on deep Q-network,” Ph.D Dissertation, School Comput. Sci. Technol., Suzhou Univ., Suzhou, China, 2017.

[28] L. Ming, Z. Jihong, and S. Zengqi, “Impact of experience replay with fixed history length on Q-learning,” Comput. Eng., vol. 32, no. 6, pp. 7–10, 2006.

[29] Z. Fei, W. Wen, L. Quan, and F. Yuchen, “A deep Q-network method based on upper confidence bound experience sampling,” J. Comput. Res. Develop., vol. 55, no. 8, pp. 1684–1705, 2018.

[30] Z. Wei, J. Xu, Y. Lan, J. Guo, and X. Cheng, “Reinforcement learning to rank with Markov decision process,” in Proc. Int. Acm SIGIR Conf. Res. Develop. Inf. Retr., 2017, pp. 945–948.

[31] T. Schaul et al., “Prioritized experience replay,” Comput. Sci., Feb. 2015. [Online]. Available: https://arxiv.org/pdf/1511.05952v1.pdf

FENgkai KE received the Ph.D. degree from the School of Mechanical Engineering and Science, Huazhong University of Science and Technology (HUST), in 2016. In 2018, he worked as a Postdoctoral Researcher with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He is currently working with the School of Mechanical Engineering, Hubei University of Technology, Wuhan, Hubei, China. During his doctoral and postdoctoral studies, he participated in a number of national and provincial natural science fund projects. He has published more than ten authoritative journals and international conference papers. His main research interests include deep learning, reinforcement learning, medical image processing, and so on.