Learning Pick to Place Objects using Self-supervised Learning with Minimal Training Resources

Pick-to-Place Objects with Self-supervised Learning

Marwan Qaid Mohammed, Lee Chung Kwek, Shing Chyi Chua
Faculty of Engineering and Technology
Multimedia University (MMU)
Melaka, Malaysia

***Abstract*—Grasping objects is a critical but challenging aspect of robotic manipulation. Recent studies have concentrated on complex architectures and large, well-labeled data sets that need extensive computing resources and time to achieve generalization capability. This paper proposes an effective grasp-to-place strategy for manipulating objects in sparse and chaotic environments. A deep Q-network, a model-free deep reinforcement learning method for robotic grasping, is employed in this paper. The proposed approach is remarkable in that it executes both fundamental object pickup and placement actions by utilizing raw RGB-D images through an explicit architecture. Therefore, it needs fewer computing processes, takes less time to complete simulation training, and generalizes effectively across different object types and scenarios. Our approach learns the policies to experience the optimal grasp point via trial-and-error. The fully conventional network is utilized to map the visual input into pixel-wise Q-value, a motion agnostic representation that reflects the grasp's orientation and pose. In a simulation experiment, a UR5 robotic arm equipped with a Parallel-jaw gripper is used to assess the proposed approach by demonstrating its effectiveness. The experimental outcomes indicate that our approach successfully grasps objects with consuming minimal time and computer resources.**

Keywords*—Self-supervised; pick-to-place; robotics; deep q-network

I. Introduction

Dexterous grasping is a crucial ability of robots that enables them to assist and substitute humans in accomplishing various tasks that might be too dangerous or tedious to do. Deep learning (DL) allows computational models composing multiple processing layers to learn data representation with multiple levels of abstraction [1]. On the other hand, Reinforcement learning (RL) relates how software agents learn to take actions in an environment such that some notion of cumulative reward is maximized via a trial-and-error approach [2]. A typical deep reinforcement learning (deep-RL) combines these two machine learning methods [3], which leverages the representation power of deep learning to solve the reinforcement learning problem. When applied to robotic grasping, the robot observes the environment through RGB-D data, and attempts an optimal action the predefined policy. Robotics can be used in nearly every circumstance, but particularly in cluttered environments, where the need for enhanced grasping efficiency demands. Object grasping is a typical robotics challenge that has made substantial progress in recent years, which is an essential step in many robotic tasks [4]. Objects removal task has been extensively researched in many studies. yet it is a challenging task in robotic manipulation [5].

The process by which a robot learns to grab and remove objects from its workstation is called object removal. Although many studies have focused on learning to grasp a single or multiple objects, some of these studies have examined how to overcome the difficulty of grasping in crowded surroundings where things seem to be stuck together in a pile. The robot must be able to detect and interpret objects and their environment in this situation, as well as effectively remove the objects from the robot's workspace. Recently, a standard Deep-RL has been used in a variety of robotic applications [6], including placement [7], grasping deformable objects [8], and grasping in a cluttered environment [5]. Meanwhile, it has advanced technology by integrating visual and tactile input, particularly in robotic grasping [4]. Additionally, deep-RL has offered great solutions for difficult-to-perform and repeat tasks via the use of end-to-end training. Since robots are usually effective at grabbing a variety of objects, interest in robots with warehouse automation skills has steadily increased in recent years. RGB-D data is increasingly being used to enhance robotic vision-based grasping in cluttered environments. Zeng et al. [9] developed a method that utilizes multi-view RGB-D data in conjunction with self-supervised and data-driven learning.

In [10], the authors utilised a straightforward view-based rendering as a forward-prediction model. To generate reliable dense visualizations of objects from RGB-D data for robotic manipulation, Florence et al. [11] presented the Dense-Object Net using a ResNet architecture. Some studies used only RGB data, obviating the necessity of depth images. The use of depth images was not required in certain studies. For example, [12] proposed GANs that could use a single batch of RGB data to predict a hand's form and location for various object grasping. Kalashnikov et al. [13] employed RL to generate a grasp pose detection dataset from RGB data in cluttered settings. The learned policies were optimized utilizing the aforementioned methods' experience. However, learning typically takes days to acquire enough experience training iterations since it needs significant computing resources to calculate the large quantity of required data. Using a large dataset [14] (e.g., recognizing

graspable poses with RGB-D data [15], point clouds[16], semantic segmentation based grasp[17]) necessitates a large amount of memory and a powerful graphics processor unit (GPU), such as NVidia Drivers, which is currently one of the difficulties in Supervised learning. In this paper, we propose an explicit pick-to-place framework that is less sophisticated than others [18]–[22] and that can be trained with appropriate CPU-Memory or GPU-Memory while taking into account training time and sufficient data for adequate evaluation analysis.

We propose a pick-to-place approach in self-supervised learning, in which RGB-D images are mapped to grasping actions through a fully connected network (FCN). The executed action is evaluated via trial-and-error by maximizing the rewards. The paper's primary contributions are as follows:

- To create an all-inclusive explicit manipulation approach that incorporates both picking and placement activities.
- To minimize the complexity of the model architecture to do training with minimal GPU or CPU resources.
- To increase the chance of robotic grasping in cluttered environment.

The paper is organized as follow: Section 2 discusses related studies, while Section 3 discusses the proposed approach's methodology in detail, including an overview of the strategic approach. The simulation experiment is given in the next section. Section 5 summarizes the findings and discusses them. The conclusion of the paper is presented in Section 6.

II. Related Work

Several studies have focused on robotic grasping, especially in dense surroundings, and proposed solutions using deep-RL, an efficient method. This area requires further investigation and understanding of the problems. Taking everything from a robot's workplace is part of cleaning up a cluttered environment. The robot must be able to perceive, interpret, and act on its surroundings and objects. When objects are physically close together, the robot's gripper must locate a place for its fingers to grasp. Zeng et al. [9] proposed to train Q-learning on FCNs. The vision system takes RGB-D images from various angles. The robot's workspace collects RGB-D images from 15 to 18 angles. Each RGB image feeds an FCN for 2D object segmentation. The final product is 3D. This data is then combined with an existing 3D model to get the 6d posture. In [13], QT-Opt, an off-policy training technique based on Q-continuous learning's action extension, is proposed. Closed-loop vision-based control is enabled via dynamic manipulation and scalable RL. The robot constantly updates its grab tactic to improve long-horizon grasp success probability.

Florence et. al [11] used the idea of self-supervised learning. The Dense-Object Network is employed, which uses the ResNet model to learn dense visual representations of objects from RGB-D data for robotic grasping in cluttered surroundings. However, it only shows a dense descriptor for three object classes, but more object classes might complicate the descriptor space segregation. Furthermore, mask region-based convolutional neural network (R-CNN) incorporates pixel-wise multi-class instance mask prediction for visible and occluded area mask segmentation. The [23] proposed learning instances and semantic segmentation for visible and occluded regions. Semantic segmentation utilizes Fully convolutional instance-aware semantic segmentation (FCIS) architecture to estimate position-sensitive masks using multi-class instance masks. It requires a dataset including all of the objects' possible occlusion states, labels, and masks; the amount of effort required to complete this task increases exponentially with the number of items. Those studies seem to be a time-consuming and complicated approach.

Active learning trained an RL framework on the intended neural network (NN) [24]. The grasp space is explored using a set of rules. Weighted retraining reduces the effect of measurement mistakes. The pixel-attentive policy gradient method proposed in [25] uses a single depth image and progressively zooms onto a specific area of the image to estimate the optimum grasp. Using Generative models to arrange multi-finger grasps is more difficult than using parallel-jaw grasps in a cluttered environment. In [26], a real-time deep convolutional encoder-decoder NN for open-loop robotic grasping has been proposed. In their method, UG-Net can estimate the quality and posture of a grip using a depth image. In [10], rendering or simulating future states concerning many possible actions is re-used. As a result, an end-to-end 6-DoF closed-loop grasping model using RL is shown employing a learned value function (Q-value). Also, an RL framework and 3D vision architectures were proposed [27] using hand-mounted RGB-D cameras. However, manipulation with more task-dependent representations must be learned from limited training data. Also, Yang and Shang [28] suggested an attention DQN for robotic grasping in clutter. Whereas, Assembly task to grasp the objects and place in stacking manner has been executed in [29].

In [12], generative adversarial networks (GANs) were introduced to estimate the hand shape and position for multiple item grasping. However, unstable training requires careful hyperparameter tuning. For 6-DoF grasping, the generative attention learning (GenerAL) method [30] has been provided, which uses deep RL to directly output the final position and configuration of the fingers. In another study, a generative grasping GG-CNN is provided [31] to extract the grip quality from a depth image. It also predicts the optimum grip based on the location, angle, and grasping breadth. However, an inaccurate grasp width estimate causes gripper collisions on large and small objects. In cluttered scenes, an end-to-end network (Contact-GraspNet) has been presented [32] to effectively and automatically distribute 6-DoF parallel-jaw grasps using depth data while preventing collisions. The limited grip breadth prevents it from grasping heavy objects. The discontinuous selection boundary makes predictions less trustworthy. Besides, The collision-aware reachability predictor (CARP) approach [33] has been proposed to learn to estimate the probabilities of a collision-free grasp position, thus substantially enhancing the grasping of objects in challenging situations. In addition, Generative deep dexterous grasping in clutters (DDGC) proposed to generate a set of collision-free multi-finger grasps in cluttered scenes. High-quality grasps produced by DDGC do not always give a successful grasp [34].

There are some challenges that arise as a consequence of the training requirements and the time required to complete grasping activities. Certain failure scenarios occur in clutter circumstances due to the clutter being so dense that there is no space for the robot to place its fingers. Additionally, it needs a simulation setup and, in most cases, an extensive parameter search to function well. As a result, it is computationally intensive, taking between tens of seconds and minutes to complete. Similarly, batch training may not be optimal for predictions involving dense clutter. Learning often involves computing a massive quantity of necessary data, which results in a high cost of training setup and a long amount of time required to acquire proficiency. As a result, the majority of studies used sophisticated architectural frameworks that need a powerful graphics processing unit (GPU) to accelerate the training process, which not every academic can afford. Additionally, they focused on executing grasp movements without regard for where the object should be placed after it was grasped. To overcome the aforementioned challenges, we propose a grasp action with a single FCN of posture estimation using an explicit approach that consumes less time while performing efficient grasping. The proposed approach's purpose is to avoid model architectural complexity to enable training to be done with minimal GPU or CPU resources. Additionally, it focuses not just on grasping but also on placing tasks. Accordingly, a complete manipulation system (pick-to-place) is developed, which is learned collaboratively via end-to-end learning.

III. Methodology

This section will explain the system's architecture and functions in detail. Then, the grasp and placement actions will be described in terms of how they perform and how they grasp and place actions' rules-based coordination works. The reinforcement learning formula and associated rewards will be explained, as well as how this component contributes to the robot's task learning.

A. Approach Overview

The proposed approach is intended to minimize the demands of the computing process, which could have an impact on the cost of time used during operation (Figure 1). The approach architecture is designed to run in a reasonable amount of time on a moderate CPU or GPU. The purpose of this paper is also to create a self-supervised learning manipulation approach that avoids the inherent complexity of approach architecture.

Firstly, the camera captures the RGB-D images, which then projected to generate the color (C_h) and depth (D_h) heightmaps. The C_h and D_h will be rotated ($\circlearrowleft \times N$) before being forwarded into the conventional layers (a 2-layer residual networks [35]). The residual networks will reduce the input parameters of DenseNet-121 to be 1024 instead 2048, which can effectively minimize the time-consuming, and run on moderate CPU and GPU. Then, the extracted features are then fed to a DenseNet-121 [36], a pre-trained model on ImageNet [37], to create motion agnostic features. Then, the motion agnostic features are used as inputs by the grasp net ϕ_g followed by bilinear upsampling, which estimates the grasp Q-maps ($Q_g\ (s_t, a_t) \in \phi_g$). A three-layer residual network is used in the ϕ_g. Finally, the robot executes the predicted best grasp, corresponding to the highest Q-value. Rewards are then assigned automatically depending on the success of grasps. The experience replay [38] is employed, which used to store the agent's experiences at each time step in a data set $e_t = (s_t, a_t, r_t, s_{t+1})$ that is pooled across many episodes to create a replay memory. Then, like with DQN, we randomly sample the memory for a minibatch of experience and utilize this to train off-policy.

B. Q-Learning and Reward Function

The representation image of the environment is viewed as a state (s_t) in this article, which is the deep network's input. The output is the action with the highest action-value, and it results in an immediate reward. As a consequence, as demonstrated in Eq. (1) [2], the policy (π) is reinforced by selecting the action with the highest state-action value. The agent's goal is to select the best action that maximizes the action-value function and the sum of future reward expectation returns. Maximization is accomplished by selecting the optimal value action (among all potential actions).

$$\pi^*(s) = arg\ max_a\ Q_\pi(s, a\ ). \quad (1)$$

To estimate the optimal grasping action, the approach is trained via Q-learning on FCN. The Q-value is learned in association with the offline policy, as in Eq. (2).

$$Q_{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q(a', s') - Q(s_t, a_t)) \quad (2)$$

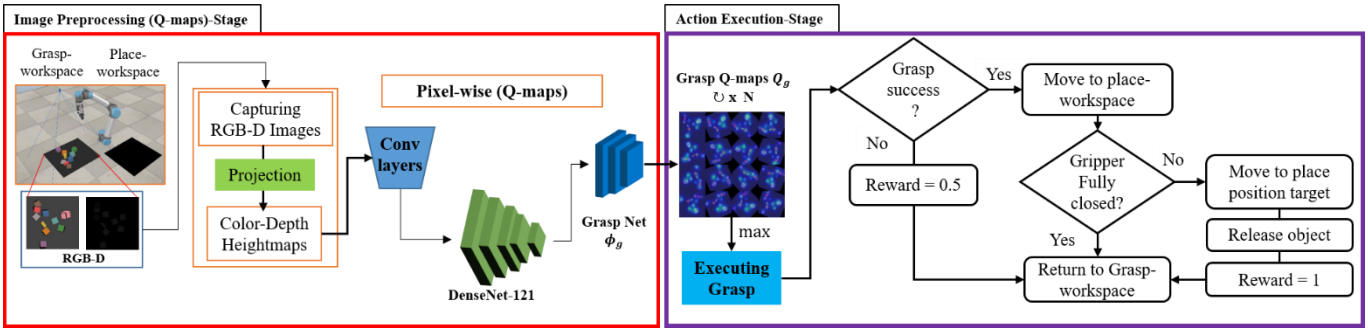

Fig. 1. The Workflow of Proposed Approach based Picking and Placing of Objects.

The $\mathcal{Q}(s_t, a_t)$ parameter represents the current Q-value, which is updated during the training, and the (α) variable represents the assigned learning rate, which is between 0 and 1. Meanwhile, The discount factor (γ) is set to 0.5. The current reward (r_t) is received by transitioning from the present state (s_t) to the future state ($s' = s_{t+1}$). The reward is required to inform the robot about which state-action pairs are efficient and which are not. The initial value of r_t is 0, but it is increased throughout the training process to stimulate the robot to perform grasp tasks and reduce the loss value. The grasp prediction yields the future reward (e.g. $\max_{a'} Q(a', s')$). Since Q-learning is trained on FCN, the learning rate is used in the stochastic gradient descent optimizer's back-propagation, it is no longer essential to include it in the Q-learning equation. After removing the learning rate, the two terms cancel out (as written in Eq. (3)).

$$r_t + \gamma \max_{a'} Q(a', s') \tag{3}$$

Accordingly, we set the reward function as follows:

- $r_g(s_t, s_{t+1}) = 0.0$ for grasp if it fails and gripper never come in contact with the objects,
- $r_g(s_t, s_{t+1}) = 0.5$ for grasp if it fails and gripper come in contact with the objects.
- $r_g(s_t, s_{t+1}) = 1.0$ for successful grasp and place the object.

C. Grasp and Place Primitive Actions

Each action (a_t) is represented as a primitive motion (Ψ) at 3D position(P), which is projected from the pixel (px) of the heightmap image that depicts the state (s_t), as shown in Eq. (4).

$$a = (\Psi, \mathrm{P}) \mid \Psi \in \{grasp\}, \mathrm{P} \longrightarrow \rho x \in s_t \tag{4}$$

A gripping action is presented as primitive motion. In one of 16 positions, the grasping motion is executed utilizing the center point within the gripper's parallel-jaw of top-down grasp. The robot moves its gripper's fingers down 3 cm of the expected location before closing its fingers to ensure that it reaches the desired object. The difference between the location of the gripper before and after gripping attempts is compared to its threshold value to detect a grasp action. The distance between the gripper's fingers and the workspace, which is 300 cm, is used as the threshold value. A successful grabbing attempt, on the other hand, is recorded when the fingers are not entirely closed, indicating that the object stays intact within the gripper's fingers until the robot places the object down.

In the next stage, once the robot has gripped an object, the placing operation will be carried out, as shown in Figure 1. The robot arrives at the workplace. The gripper's state is then verified to make sure the object is still in position. The placement process will be interrupted if there is slippage. After then, the robot returns to its starting location for a new iteration. For example, the robot will then place the object into the pre-defined place-workspace if it has been successfully grabbed. If the robot's gripper is not fully closed during a placement job, it implies that the object is within the gripper's fingers, allowing the robot to continue placing the object; otherwise, the robot will interrupt and resume grasping instead of placing the object.

IV. Simulation Experiment

In this paper, V-REP [39] is used to simulate an experiment using a UR5 robot equipped with a parallel jaw gripper. The robot uses an RGB-D camera to observe its environment. The color and depth images are captured at a 640 by 480-pixel resolution. A 3.7 GHz Intel Core i7-8700HQ CPU and an NVIDIA 1660Ti GPU power the PyTorch-based prediction network.

A. Training Session

Self-supervised learning using a simulation platform is used to train the proposed approach. A similar training procedure to that described in [40] is used. For training, a collection of ten objects of different shapes is randomly placed into the robot's workspace. By trial and error, the robot learns to perform picking and a placement action. After clearing the workspace of all objects, another set of 10 objects is dropped for additional training. Continuous data collection occurs until the robot has completed 3K training iterations.

B. Testing Session

We conducted a series of experiments to determine if the proposed approach is successful at accomplishing the grasping-to-placing task. We validate our approach using scenarios involving randomly cluttered objects with varying degrees of clutter, namely spared, medium, and dense clutter levels, as shown in Figure 2.

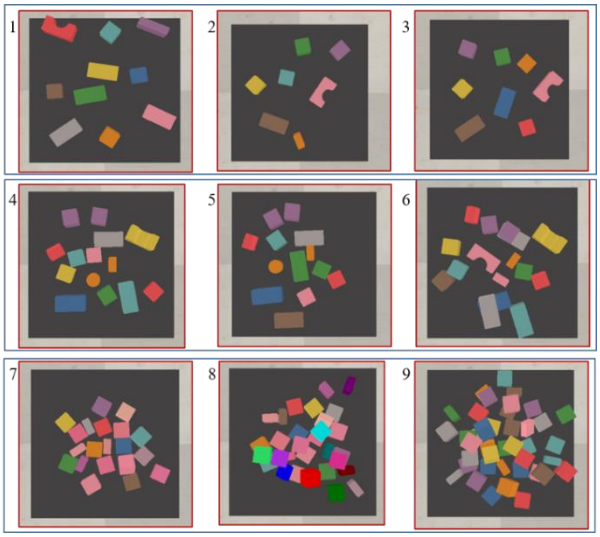

Fig. 2. A Series of Randomly Sparsely, Medium, and Densely Cluttered Object Challenge Scenarios.

- Sparsely cluttered objects scenario (test-cases 1-3): the objects are randomly distributed on the workspace in groups of 6–7.
- Medium cluttered objects scenarios (test-cases 4-6): The objects disperse in a random order of 9–10 objects that are distributed in close contact with one another. They are more challenging to perform than the first type of scenario.

- Densely cluttered objects scenarios (test-cases 20-30): Three scenarios, including a random selection of 20–30 objects, conduct to assess the proposed approach, which implies more challenging than the previous two sorts of scenarios.

C. Evaluation Metrics

The proposed approach is assessed using the test scenarios described before. The robot must retrieve and clean all objects from the workspace in order to place them into place-workspace. Five test runs (denoted by n) are conducted for each test case. The workspace contains between 6 to 40 objects. Three assessment metrics are utilized to evaluate the models' performance. The greater the value for each of these metrics, the better. These are the metrics.

- The grasp success rate: Ratio of the success grasp attempts to the total of executed actions over n test runs per each test case, and
- The place success rate: ratio of the number of successful place over the number of successful grasp through whole run tests of each case test.
- The completion rate: It's the average of the total number of completed objects divided by the total number of objects. It is used to measure the capability of proposed approach to grasp all objects in each test case without failing for more than five actions consecutively.

V. Result and Discussion

This section organizes the findings into training and testing sessions. The proposed method's results will be shown throughout the training session via graphs of grasp success rate, which illustrate how the proposed approach performed during the training stage and how fast and effectively it learned. The testing session consists of a sequence of test cases, each of which is conducted five times. The models' performance is assessed using their grasp success, place success and completion rates.

A. Training Session Outcome

The proposed approach (PA) was trained alongside other baselines utilizing a different training procedure. The grasping performance is evaluated by the proportion of successful grasp attempts made within the last 200 tries (m = 200). The percentage is scaled by a factor of i/m in the earlier training trials, i.e., trials $i < m$. Figure 3 illustrates the grasp success rate graphs for 4000 training iterations. In this section, we trained the suggested method using different variables to evaluate whether or not these aspects affect the grasping performance when taken into account.

1) PA-nodepth: the proposed approach is trained only on color image data, ignoring depth information. It can be shown that when depth is not included during training, it affects grasp performance, with a grasp rate of almost 73%. Additionally, it requires many trials at the beginning of learning to gain expertise with the environment to boost its performance.

2) PA-nopretrain: the proposed approach leverages the use of the DenseNet-121 model, which was pre-trained on ImageNet. However, we need to evaluate our proposed approach's effectiveness in the absence of ImageNet pre-training. The training session findings show that pre-trained models assist the learning process by improving grasping performance with a minimum number of iterations, in comparison to grasping performance when no pre-training model was used, which struggled for the first 200 iterations of total training iterations, within the range of 65% to 70% grasp success rate.

3) PA-noER: In this portion, the proposed approach was trained without using experience replay (ER), which stores the agent's experiences at each time step for use as an off-line policy in subsequent training iterations. The success rate graph indicates that ER has an effect on learning, gradually improving grasping skill in comparison to other factors. The first 500 iterations of a training session achieve a success rate of almost 50%, indicating that the model could be significantly influenced by no experience replay.

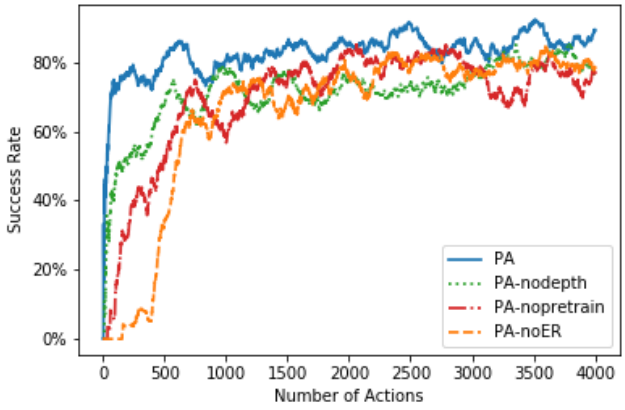

Fig. 3. The Proposed Approach's Performance in Comparison to other Baseline Models in Terms of Grasp Success Rate throughout Training Sessions.

The proposed approach, when combined with pretrain, ER, and RGB-D data, has been demonstrated to significantly improve grasping performance with a success rate of almost 83 % and steady learning throughout the training. In term of time-consuming, each iteration takes an average 4 second on the GeForce GTX-1660Ti (6GB) and GeForce GTX 1650 Ti (4GB). We also test the time consuming of the proposed approach on the CPU with RAM of DDR4 (16GB) with 30 seconds. The whole training session for each baseline it takes almost 4-4.5 hours.

B. Testing Session Outcome

The grasp success, place success, and completion rate are the two evaluation metrics used to assess the performance of the proposed approach. The proposed approach is tested using three scenarios: 1) Sparsely clutter objects, 2) medium clutter objects and 3) densely clutter objects (Figure 2). These type of scenarios are varied in level of clutter challenge with range of objects 6-40 objects.

In Table 1, the results indicate that our approach performed well in more difficult tasks, especially in the first two types of

situations, namely sparsely and moderately cluttered objects, where it achieved a grasp success rate of 93.2 % and 86.1 %, respectively. However, performance degrades as the test scenario becomes more cluttered, with a grasp success rate of 71.7 %. In general, the proposed approach is capable of effectively performing grasping tasks, with a completion rate of about 95% in all scenarios. It implies that it is capable of efficiently moving objects from the robot's workspace to the place workspace.

When we compare our approach to others, many factors must be addressed, as shown in Table II. Interestingly, our approach is capable of grasping with a minimum of time and training resources. In comparison, other approaches need a minimum of ten seconds to complete one iteration. Similarly, if their approaches are carried out on the CPU, they may take multiple minutes to finish a single iteration due to the complexity of the computing process. On the other hand, our approach is capable of performing grasping tasks on the CPU as well, with each iteration averaging 30 seconds.

TABLE I. ASSESSMENT OF RANDOMLY CLUTTERED OBJECT CHALLENGE SCENARIOS

Metrics (average %)	Test scenarios		
	Sparsely Clutter	***Medium Clutter***	***Densely Clutter***
Grasp Success Rate	93.2	86.1	71.7
Place Success Rate	100	95.7	93.8
Completion Rate	100	100	86.1

TABLE II. COMPARISON OF THE PROPOSED APPROACH WITH OTHERS

Method	Training Resources				Time-Consuming	Grasp Success %	Execution action	
	CPU	***GPU***						
	RAM-16GB	***4GB***	***6GB***	***≥ 8GB***			***Grasp***	***Place***
[9]	×	×	×	✓	8-15 seconds	66.7%	✓	×
[18]	×	×	×	✓	10-18 seconds	78%	✓	×
[29]	×	×	×	✓	15-20 Seconds	81.2%	✓	✓
[28]	×	×	×	✓	7-10 seconds	73.5%	✓	×
Ours	✓	✓	✓	✓	4-5 seconds	83.7	✓	✓

VI. CONCLUSION AND FUTURE WORK

One of the difficulties faced by robots is performing grasping tasks in an unstructured environment. In this paper, the proposed approach, which is based on DQN, showed exceptional grasping performance in a range of test scenarios including randomly cluttered objects. The proposed method has been proven its capability of removing objects from a workspace efficiently. The approach achieves an 83.1 % grasp success rate in cluttered object settings, demonstrating that it is capable of successfully performing a grasping challenge. Additionally, even in challenging circumstances, the proposed approach obtains a high completion rate (96.1 % in all cluttered environment scenarios). In terms of time required, each iteration takes an average of four seconds on the GPU and 30 seconds on the CPU. Significantly, the proposed learning approach proved successful in addressing the aforementioned problems, namely the time and training resources requirements. On the other hand, the proposed approach becomes inefficient as the number of objects increases. This deficiency could well be addressed in the future via the potential merging of grasp and push. Similarly, simulations have been used to assess the proposed approach, which is another possible disadvantage to consider. However, the proposed approach has been evaluated only via simulations, which is a possible drawback to consider. The proposed approach will be implemented on hardware in a future study, giving strong validation for those interested in doing further research.

ACKNOWLEDGMENT

Authors are thankful to Multimedia University (MMU) for supporting this research. This research is supported by Multimedia University (MMU) through MMU GRA Scheme (MMUI/190004.02.) and MMU Internal Fund (MMUI/210111).

REFERENCES

[1] H. Il Suk, “An Introduction to Neural Networks and Deep Learning,” in Deep Learning for Medical Image Analysis, 1st ed., Elsevier Inc., 2017, pp. 3–24.

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. MIT press, 2018.

[3] V. François-lavet et al., “An Introduction to Deep Reinforcement Learning,” Found. Trends® Mach. Learn., vol. 11, no. 3–4, pp. 219–354, 2018.

[4] Q. M. Marwan, S. C. Chua, and L. C. Kwek, “Comprehensive Review on Reaching and Grasping of Objects in Robotics,” Robotica, vol. 39, no. 10, pp. 1849–1882, 2021.

[5] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, “Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations,” IEEE Access, vol. 8, pp. 178450–178481, 2020.

[6] J. Andrew Bagnell, “Reinforcement Learning in Robotics: A Survey,” Springer Tracts Adv. Robot., vol. 97, pp. 9–67, 2014.

[7] W. Guo, C. Wang, Y. Fu, and F. Zha, “Deep Reinforcement Learning Algorithm for Object Placement Tasks with Manipulator,” in 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR), 2018, pp. 608–613.

[8] H. Han, G. Paul, and T. Matsubara, “Model-based reinforcement learning approach for deformable linear object manipulation,” in 2017 13th IEEE Conference on Automation Science and Engineering (CASE), 2017, pp. 750–755.

[9] A. Zeng et al., “Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1383–1386.

[10] S. Song, A. Zeng, J. Lee, and T. Funkhouser, “Grasping in the Wild: Learning 6DoF Closed-Loop Grasping From Low-Cost

Demonstrations," IEEE Robot. Autom. Lett., vol. 5, no. 3, pp. 4978–4985, 2020.

[11] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation," arXiv:1806.08756v2, pp. 1–12, 2018.

[12] E. Corona, A. Pumarola, G. Alenyà, F. Moreno-Noguer, and G. Rogez, "GanHand: Predicting Human Grasp Affordances in Multi-Object Scenes," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5030–5040.

[13] D. Kalashnikov et al., "QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation," arXiv:1806.10293v3, no. L, pp. 1–23, 2018.

[14] J. Mahler et al., "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics," arXiv Prepr. arXiv1703.09312 v3, pp. 1–12, 2017.

[15] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," Int. J. Rob. Res., vol. 34, no. 4–5, pp. 705–724, 2015.

[16] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," Int. J. Rob. Res., vol. 37, no. 4–5, pp. 421–436, Apr. 2018.

[17] M. Q. Mohammed, L. C. Kwek, S. C. Chua, and E. A. Alandoli, "Color Matching Based Approach for Robotic Grasping," in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), 2021, pp. 1–8.

[18] J. Mahler and K. Goldberg, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," in Proceedings of the 1st Annual Conference on Robot Learning, 2017, vol. 78, pp. 515–524.

[19] M. Q. Mohammed, M. F. Miskon, M. B. Bin Bahar, and S. A. Ali, "Comparative study between quintic and cubic polynomial equations based walking trajectory of exoskeleton system," Int. J. Mech. Mechatronics Eng., vol. 17, no. 4, pp. 43–51, 2017.

[20] A. Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in IEEE International Conference on Robotics and Automation (ICRA), 2019, pp. 3750–3757.

[21] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, "Pick and Place Objects in a Cluttered Scene Using Deep Reinforcement Learning," Int. J. Mech. Mechatronics Eng. IJMME, vol. 20, no. 04, pp. 50–57, 2020.

[22] S. A. Ali, M. Fahmi Miskon, A. Zaki Hj Shukor, and M. Qaid Mhoammed, "The Effect of Parameters Variation on Bilateral Controller," Int. J. Power Electron. Drive Syst., vol. 9, no. 2, p. 648, 2018.

[23] K. Wada, K. Okada, and M. Inaba, "Joint learning of instance and semantic segmentation for robotic pick-and-place with heavy occlusions in clutter," in Proceedings - IEEE International Conference on Robotics and Automation, 2019, vol. 2019-May, pp. 9558–9564.

[24] L. Berscheid, T. Rühr, and T. Kröger, "Improving Data Efficiency of Self-supervised Learning for Robotic Grasping," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 2125–2131.

[25] B. Wu, I. Akinola, and P. K. Allen, "Pixel-Attentive Policy Gradient for Multi-Fingered Grasping in Cluttered Scenes," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 1789–1796.

[26] Y. Song, Y. Fei, C. Cheng, X. Li, and C. Yu, "UG-Net for Robotic Grasping using Only Depth Image," in 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2019, pp. 913–918.

[27] X. Chen et al., "Transferable Active Grasping and Real Embodied Dataset," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3611–3618.

[28] Z. Yang and H. Shang, "Robotic pushing and grasping knowledge learning via attention deep Q-learning network," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12274 LNAI. Academy for Engineering and Technology, Fudan University, Shanghai, China, pp. 223–234, 2020.

[29] K. Zakka, A. Zeng, J. Lee, and S. Song, "Form2Fit: Learning Shape Priors for Generalizable Assembly from Disassembly," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9404–9410.

[30] B. Wu et al., "Generative Attention Learning: a ``GenerAL{''} framework for high-performance multi-fingered grasping in clutter," Auton. Robots, vol. 44, no. 6, pp. 971–990, Jul. 2020.

[31] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," Int. J. Rob. Res., vol. 39, no. 2–3, pp. 183–201, 2020.

[32] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes," arXiv:2103.14127v1, pp. 1–7, 2021.

[33] X. Lou, Y. Yang, and C. Choi, "Collision-Aware Target-Driven Object Grasping in Constrained Environments," arXiv:2104.00776v1, pp. 1–7, 2021.

[34] J. Lundell, F. Verdoja, and V. Kyrki, "DDGC: Generative Deep Dexterous Grasping in Clutter," IEEE Robot. Autom. Lett., vol. 6, no. 4, pp. 6899–6906, 2021.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[36] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), 2017, pp. 2261–2269.

[37] L. Fei-Fei, J. Deng, and K. Li, "ImageNet: Constructing a large-scale image database," in IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[38] M. Andrychowicz et al., "Hindsight experience replay," in Advances in Neural Information Processing Systems, 2017, vol. 2017-Decem, pp. 5049–5059.

[39] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in IEEE International Conference on Intelligent Robots and Systems, 2013, pp. 1321–1326.

[40] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning Synergies Between Pushing and Grasping with Self-Supervised Deep Reinforcement Learning," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 4238–4245.