Enhanced Deep Reinforcement Learning for Robotic Manipulation: Tackling Dynamic Weight in Noodle Grasping Task

Prem Gamolped, Ja Sin Yon Pang, Vjosa Bytyqi, Eiji Hayashi
Department of Creative Informatics, Kyushu Institute of Technology,
680-4 Kawazu, Iizuka, Fukuoka, 820-8502, Japan
Email: gmp.prem1997@gmail.com, yon-pang.sin-ja413@mail.kyutech.jp,
vjosabytyci@gmail.com, haya@mse.kyutech.ac.jp

Abstract

Handling food items with dynamic weight changes over time, which alter physical properties such as shape, size, and weight, poses significant challenges, particularly when precise output weight is required. This study introduces an enhanced deep reinforcement learning framework for robotic manipulation, focusing on the task of spaghetti grasping. Building on prior research, we propose a data augmentation strategy that simulates diverse environmental conditions, including variations in image observations and the physical properties of spaghetti, to improve models. The model is validated using metrics such as grasp success rate, average grasp time, and generalization score under varying environmental conditions. This work advances the robustness of robotic models in previously unseen environments.

Keywords: Robotic Manipulation, Deep Reinforcement Learning (DRL), Data Augmentation, Grasping Task

1. Introduction

The automation of complex food handling tasks has emerged as a critical focus in robotics, addressing the growing demand for precision and efficiency in food production lines. Building on our previous research [1] in robotic manipulation for assembling bento boxes, this study extends the exploration of deep reinforcement learning (DRL) techniques to address the challenges of handling food items with dynamic weight changes. While our earlier work focused on grasping and placing spaghetti with a specific weight under a controlled environment, this study will tackle the broader problem: how robotic systems can adapt to dynamic changes in the physical properties of food items, such as weight, shape, and size, over time. Handling deformable food items, such as spaghetti, presents unique challenges in robotic manipulation. Spaghetti's properties are affected by factors such as humidity, temperature, and preparation methods—which necessitate precise grasping strategies to achieve consistent grasping weight. In our previous study, a DRL-based robotic system demonstrated significant success in grasping spaghetti with a Mean Absolute Percentage Error (MAPE) of approximately 19%. This success defines the potential of DRL-based in tackling complex food-handling tasks but also reveals the need for more robust solutions capable of adapting to environmental variations.

This study introduced an enhanced DRL framework specifically designed for dynamic food-handling scenarios. The framework incorporates a data augmentation strategy to simulate diverse environmental conditions, enabling the robotic system to generalize under diverse environmental conditions. By integrating these data augmentation techniques to make the environment more varied in the training process, the proposed aims to improve the robustness and adaptability of DRL models in real-world environments. In this study, we will use the metric to assess the model's performance in dynamic settings. These include grasp success rate (GSR), average grasp time (AVT), task efficiency (TE), and generalization score (GS) under varying conditions. Such metrics can provide a sufficient evaluation framework, highlighting the system's ability to adapt and perform reliably despite the fluctuations in the target item's properties. The experimental validation of our approach builds upon the methodology and setup outlined in our prior work, which included a custom-designed gripper, an RGB-D camera for visual input, and a digital scale for weight measurement. Extending the prior system, the current study incorporates data augmentation [2], [3] techniques, leveraging diverse datasets to enhance the DRL model's learning process. The result is a system capable of achieving consistent grasping performance even when confronted with previously unseen environments.

2. Related Works

This section reviews previous studies in robotic food manipulation, and deep reinforcement learning (DRL) applications in dynamic environments. These provide the foundation for our proposed framework and highlight the gaps it seeks to address.

2.1. *Robotic Manipulation in Food Handling*

The integration of robotics into food processing lines has seen significant advancements over the past decade, with applications ranging from ingredient sorting to packaging. However, handling deformable objects [4], such as spaghetti, dough, or fruits, poses unique challenges due to their dynamic and unpredictable properties. Research has investigated diverse grippers and sensing technologies to enable precise handling, but achieving generalization across environmental conditions and object variability remains a significant challenge.

In a previous study [1], a DRL-based robotic system for spaghetti grasping in bento box assembly achieved a weight accuracy MAPE of around 19% using a custom gripper, RGB-D sensing, and a digital scale. While effective in controlled settings, it struggled with variations in spaghetti texture, weight, and humidity, highlighting the need for better generalization strategies to handle dynamic food properties.

2.2. *Deep Reinforcement Learning for Robotic Manipulation*

Deep Reinforcement Learning (DRL) has shown significant success in training robots to perform complex tasks in unstructured environments. Algorithms such as Soft Actor-Critic (SAC), Proximal Policy Optimization (PPO) [10], and Deep Q-Networks (DQN) [9] have been widely adopted for robotic manipulation tasks, leveraging the ability to learn from trial-and-error interactions with the environment.

Several studies have applied DRL to robotic grasping tasks, focusing on both rigid and deformable objects. For instance, this study [5] proposed a hybrid approach combining DRL and soft grippers for grasping objects of varying shapes and sizes, achieving robust performance in cluttered environments. Similarly, another study [6]. Demonstrated the potential of DRL in dexterous manipulation tasks, showcasing the ability to handle objects with complex geometries. Although these studies highlight the efficacy of DRL in grasping tasks, they largely assume static object properties, reducing their relevance for dynamic scenarios like food handling. In the context of food manipulation, similar work [7] employed reinforcement learning to pick and place cluttered objects using dense object descriptors. However, the study did not address challenges related to changes in object properties over time. This work extends these approaches by incorporating data augmentation strategies that simulate variations in environmental and object conditions, enabling the model to adapt to dynamic changes in food properties.

2.3. *Gaps and Opportunities*

Despite significant advancements in robotic manipulation, several gaps remain in the field. First, existing research predominantly focuses on static or semi-static environments, overlooking the dynamic changes in object properties observed in real-world scenarios. Second, while DRL has shown promise in robotic manipulation, its application to tasks involving deformable objects with unpredictable properties remains underexplored. Third, the evaluation metrics used in prior studies are often limited to success rates, neglecting broader measures of robustness and adaptability.

To address these gaps, this study proposes an enhanced DRL framework incorporating a data augmentation strategy to handle dynamic food items. By simulating diverse environmental conditions during training, the proposed approach aims to improve model robustness and adaptability, enabling reliable performance in previously unseen environments. Additionally, it has been defined comprehensive evaluation metrics, including grasp success rate, average grasp time, and generalization score, to assess the framework's effectiveness in dynamic scenarios.

3. Methodology

This section explains the methodology of this study, starting from the system overview, deep reinforcement learning framework, reward function design, data augmentation pipeline, and training environment setup.

3.1. *System Overview*

The robotic system in this study is based on the setup described in our previous work, which includes a robotic arm equipped with a custom gripper, an RGB-D camera for visual observations, and a digital scale for measuring grasped weights. The system was designed to grasp and manipulate spaghetti from a stationary tray, to achieve a target weight. The training and evaluation were conducted in a controlled environment, leveraging Deep Reinforcement Learning (DRL) for task optimization. The system as shown in Fig. 1 utilized a Yaskawa SIA5F robotic arm for its seven-axis versatility, coupled with a custom 3D-printed gripper. An Azure Kinect RGB-D camera provided both color and depth information, while the digital scale recorded the weight of the grasped spaghetti in real-time.

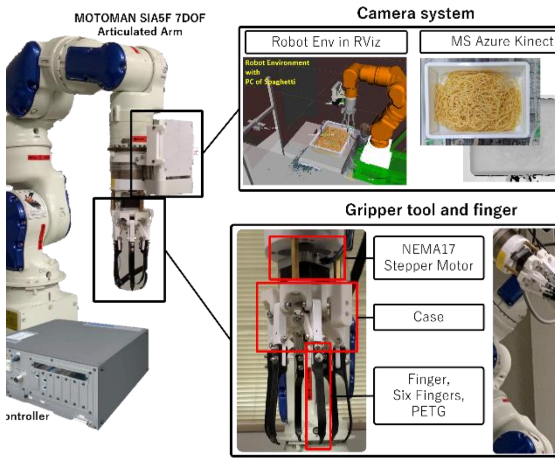

Fig.1 The robotic system in the study mainly includes an arm manipulator, gripper, and camera.

The system's overall architecture allowed precise grasping and manipulation of deformable food items like spaghetti. The PC configuration we utilized for this study is as follows:

- Ubuntu Linux 20.04 LTS
- 12th Gen Intel® Core™ i9-12900K×24
- NVIDIA GeForce RTX 3090Ti 24GB
- 64GB of RAM
- ROS Noetic
- Stable Baselines3 1.8.0 and OpenAI Gym 0.21.0

3.2. *Deep Reinforcement Learning Framework*

The Deep Reinforcement Learning (DRL) framework employed in this study builds upon the approach outlined in our prior work. The Soft Actor-Critic (SAC) [8] algorithm was utilized, an off-policy method suitable for continuous action spaces, enabling efficient training through exploration and exploitation. The policy network

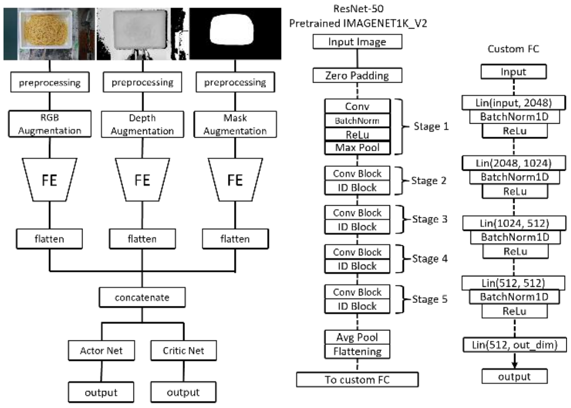

Fig.2 Net architecture for the feature extractor built on ResNet-50 with customized FC after the FE

architecture integrated a pretrained ResNet50 [11] model for feature extraction from RGB-D inputs, followed by a fully connected neural network for action generation as shown in Fig. 2. The action and observation spaces Tables 1 and 2 have been improved compared to the previous study. After obtaining the output action from the

Table 1. The observation space of this study

No.	Mass matrix	Min	Max	unit
1	RGB			pixels
2	Depth	0	255	pixels
3	Mask			pixels
4	Actions	Refer to Table 2		
5	Noodle grasp weight	0.0	200.0	grams
6	Tray weight	0.0	1200.	grams

Table 2. The action space of this study

No.	Mass matrix	Min	Max	unit
1	Pixel i			-
2	Pixel j	-1.0	1.0	-
3	Insert depth			-
4	Gripper width			-

policy, we map it to the range required by the real hardware. For the gripper width, a simple mapping is applied from the original range to the operational range of 0–50 mm. To identify the grasping point, the action parameters corresponding to pixels i and j are first mapped to the pixel range. Using the depth information of the specified pixel (i, j), the distance is calculated. The pinhole camera model is then utilized to determine the grasping point in the robot's coordinate system. For the insert depth action, this value is mapped from the predefined lowest point to the depth of the pixel (i, j).

3.3. *Reward Function Design*

The reward function utilized in this study was derived from our prior research and optimized for weight-sensitive grasping tasks. It was based on a Probability Density Function (PDF), which penalized deviations from the target weight and rewarded as in Eq. (1).

$$r(x) = a \cdot \left[\frac{1}{\sigma\sqrt{2\pi}}\right] \cdot e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]}, -a \le r(x) \le a \quad (1)$$

Here, x represents the grasped weight, μ is the target weight, and σ is the acceptable deviation. The scaling factor a ensures the reward remains within a predefined range. Additional penalties are applied for collisions or failed grasps, encouraging efficient and precise actions. The function incentivized actions resulting in weights within a predefined range, penalizing grasps that exceeded or fell short of the target weight.

3.4. *Data Augmentation Pipeline*

To improve the robustness and generalization of the robotic manipulation model, a data augmentation pipeline is implemented, focusing on RGB images, depth maps, object masks, and tray weight. For RGB augmentation, variations in brightness, contrast, saturation, and hue are applied to simulate diverse lighting conditions and environmental changes during robotic operation. Depth augmentation introduces noise and small perturbations to emulate sensor inaccuracies and real-world inconsistencies in depth perception. Object mask augmentation involves adding noise to the edges, resizing, or slightly shifting the masks to account for inaccuracies in object detection, ensuring the model can handle imperfect segmentation outputs. Tray weight augmentation introduces variations to simulate different loading scenarios, teaching the model to adapt to varying weight distributions during grasping and placement tasks.

Each augmented sample is carefully synchronized across RGB, depth, and mask channels to maintain data consistency. These augmentations not only mimic real-world variabilities but also enhance the model's ability to learn robust grasping strategies under diverse scenarios. By applying this pipeline during training, the system gains improved resilience to sensor noise, lighting changes, and variations in object properties, ultimately enabling more reliable performance in real-world manipulation tasks. This is depicted in Fig. 3.

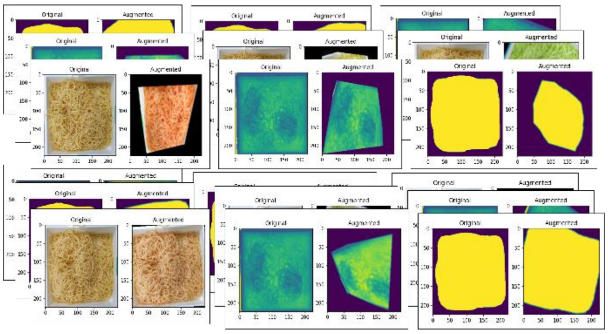

Fig.3 Data augmentation for the RGB, depth, and mask image for the observation

3.5. *Training Environment Setup*

The environment integrates RGB and depth data captured from a calibrated camera setup, along with object masks generated through segmentation techniques. These inputs are used to construct a 3D representation of the workspace, enabling precise action planning. The simulated workspace includes a tray placed within the robot's reachable area, where the robot performs grasping and placement tasks. The tray is equipped with a digital scale to measure the accumulated weight of the grasped spaghetti, providing a critical feedback signal for reinforcement learning. The robotic system operates within predefined constraints, including the gripper's range of motion, width, and insertion depth. These constraints are reflected in the action space, which maps policy outputs to real-world parameters using transformations based on the robot's kinematics and the pinhole camera model. This is depicted in Fig. 4.

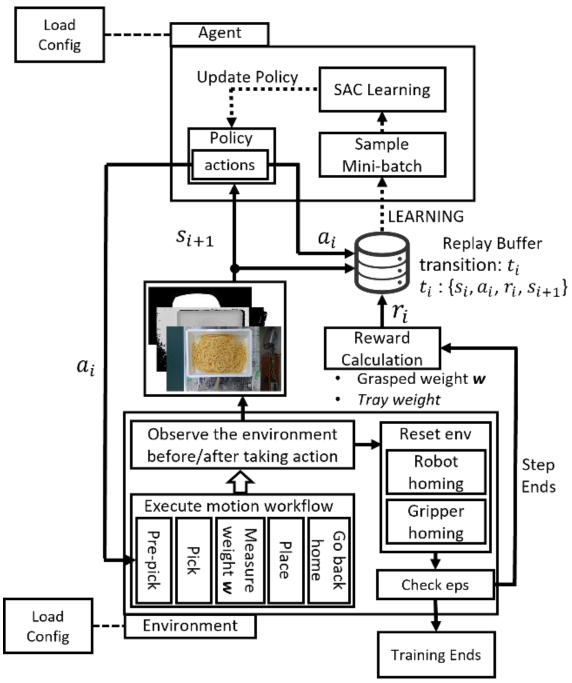

Fig.4 Diagram of how the system trains under the MDP-based method

The training environment also incorporates domain randomization to enhance generalization, varying properties such as lighting conditions, object appearance, and noise in the depth data. This ensures that the model learns robust policies that can handle real-world variabilities. The reward function is carefully crafted to balance weight accuracy, grasp stability, and execution efficiency, with penalties for dropping objects or exceeding weight limits. The environment supports episodic learning, resetting the state after each completed task or failure, and logging detailed metrics to monitor the model's performance. By combining realistic sensor inputs, diverse variations, and task-specific constraints, the training environment provides a robust foundation for learning effective policies for spaghetti manipulation in real-world scenarios.

4. Results and Discussion

4.1. *Experiment Result*

The experimental results demonstrate the effectiveness of the proposed system for autonomous robotic manipulation of deformable objects. The trained model achieved a grasp success rate (GSR) of 82%, indicating a high level of accuracy in successfully executing grasping tasks. The average grasp time (AGT) was measured at 16.99 seconds, reflecting the time efficiency of the robotic system in completing each grasp. The task efficiency (TE) was calculated as 0.0411 tasks per second, showcasing the throughput of the system. Additionally, the model was evaluated for its generalization capabilities, achieving a generalization score (GS) of 75% on new, unseen test data. This score suggests that the system can effectively adapt to new scenarios and conditions, further highlighting its robustness in real-world applications involving deformable objects like spaghetti.

4.2. *Analysis of Limitations*

Despite the promising performance of the proposed system, certain limitations were observed during the experiments, particularly when dealing with highly tangled spaghetti. In such cases, while the system is sometimes able to grasp the correct weight of the spaghetti, the tangling of the strands often interferes with the robot's ability to execute an accurate grasp. The tangles can cause the gripper to miss the target or improperly grasp multiple strands, reducing the overall grasp success rate. Additionally, the model's performance may be affected by the complexity of the entanglement, as the presence of overlapping strands introduces uncertainties that are not always accounted for in the training data. These challenges highlight the need for further improvements in the model's ability to handle highly deformable and tangled objects, which is crucial for ensuring consistent performance in real-world scenarios.

5. Conclusion

This study presents an autonomous robotic system for manipulating deformable objects, focusing on spaghetti using deep reinforcement learning (DRL). The model achieved a grasp success rate (GSR) of 82%, an average grasp time (AGT) of 16.99 seconds, and a task efficiency (TE) of 0.0411 tasks per second, demonstrating efficient performance. A generalization score of 75% indicates its adaptability to unseen scenarios. However, limitations arose when dealing with tangled spaghetti. While the system sometimes grasped the correct weight, entanglement often led to misgrasping or difficulty handling multiple strands. This issue highlights the challenge of working with tangled deformable objects.

Future work should focus on improving entanglement handling through better vision systems, real-time feedback, and expanded training data. In conclusion, while the system shows promise, overcoming tangling challenges is crucial for real-world deployment.

References

1. Gamolped, P., Koomklang, N., & Hayashi, E. (2024, February). Autonomous Spaghetti Grasping for Bento Box Assembly: A Deep Reinforcement Learning Approach. In 2024 1st International Conference on Robotics, Engineering, Science, and Technology (RESTCON) (pp. 181-186). IEEE.
2. Hu, J., Jiang, Y., & Weng, P. (2024). Revisiting Data Augmentation in Deep Reinforcement Learning. arXiv preprint arXiv:2402.12181.
3. Ma, G., Wang, Z., Yuan, Z., Wang, X., Yuan, B., & Tao, D. (2022). A comprehensive survey of data augmentation in visual reinforcement learning. arXiv preprint arXiv:2210.04561.
4. Endo, G., & Otomo, N. Development of a Food Handling Gripper Considering an Appetizing. In Book name IEEE International Conference on Robotics and Automation (ICRA).
5. F. Liu, F. Sun, B. Fang, X. Li, S. Sun, and H. Liu, Hybrid robotic grasping with a soft multimodal gripper and a deep multistage learning scheme', IEEE Transactions on Robotics, 2023.
6. A. Rajeswaran et al., Learning complex dexterous manipulation with deep reinforcement learning and demonstrations', arXiv preprint arXiv:1709. 10087, 2017.
7. H.-G. Cao, W. Zeng, and I.-C. Wu, 'Reinforcement learning for picking cluttered general objects with dense object descriptors', in 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 6358–6364.
8. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In the International conference on machine learning (pp. 1861-1870). PMLR.
9. Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020, July). A theoretical analysis of deep Q-learning. In Learning for dynamics and control (pp. 486-489). PMLR.
10. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
11. Koonce, B., & Koonce, B. (2021). ResNet 50. Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, 63-72.

Authors Introduction

Prem Gamolped

He received a bachelor's degree in engineering in 2021 from Robotics and Automation Engineering, King Mongkut's University of Technology North Bangkok in Thailand, and received a master's in 2022 from Kyushu Institute of Technology at Hayashi Laboratory. He is currently a PhD student at Kyushu Institute of Technology at Hayashi Laboratory and researches Robotics and AI.

Yon Pang Ja Sin

She obtained her bachelor's degree in mechanical engineering in 2021 from the Faculty of Engineering and Technology, Multimedia University, Malaysia. Currently, she is pursuing her master's degree at the Kyushu Institute of Technology, Japan, and is conducting research at the Hayashi Laboratory.

Vjosa Bytyqi

She received her bachelor's degree in computer engineering in 2023 from the University of Prishtina, Department of Electric and Computer Engineering. She is currently a research student at Kyushu Institute of Technology and conducts research at Hayashi Laboratory.

Prof. Eiji Hayashi

Prof. Eiji Hayashi is a professor in the Department of Intelligent and Control Systems at Kyushu Institute of Technology. He received the Ph.D. (Dr. Eng.) degree from Waseda University in 1996. His research interests include Intelligent mechanics, Mechanical systems, and Perceptual information processing. He is a member of The Institute of Electrical and Electronics Engineers (IEEE) and The Japan Society of Mechanical Engineers (JSME).