Reinforcement Learning-based Grasping via One-Shot Affordance Localization and Zero-Shot Contrastive Language–Image Learning

Xiang Long*, Luke Beddow*, Denis Hadjivelichkov,
Andromachi Maria Delfaki, Helge Wurdemann, and Dimitrios Kanoulas

***Abstract*— We present a novel robotic grasping system using a caging-style gripper, that combines one-shot affordance localization and zero-shot object identification. We demonstrate an integrated system requiring minimal prior knowledge, focusing on flexible few-shot object agnostic approaches. For grasping a novel target object, we use as input the color and depth of the scene, an image of an object affordance similar to the target object, and an up to three-word text prompt describing the target object. We demonstrate the system using real-world grasping of objects from the YCB benchmark set, with four distractor objects cluttering the scene. Overall, our pipeline has a success rate of the affordance localization of 96%, object identification of 62.5%, and grasping of 72%. Videos are on the project website: `https://sites.google.com/view/rl-affcorrs-grasp`.**

I. INTRODUCTION

Autonomous object grasping is an important and heavily studied problem in the robotics community. The developments in machine learning and artificial intelligent, in general, allowed for highly efficient methods when grasping objects for the purpose of pick-and-place [1], [2], [3]. The problem still remains open, not only when dealing with sensitive objects such as fruits, but also when the environment semantics of multi-modal data [4], [5] play a role during manipulation. In this work, we aim at dealing with these latter open questions, especially aiming for minimal prior knowledge in the presence of object and position uncertainties.

In particular, we investigate object grasping through a pipeline built around object flexibility and few-shot image methods. We detect unseen objects via a text-based grasping task description [6] and identify graspable areas on these objects via one-shot affordance localization [7], [8]. The desired object is extracted from a cluttered scene given only a support image and a brief text description, for example "chocolate pudding box". The support image includes an object similar to the target one, on which one-shot affordances are generated and matched to similar objects and regions in the novel target image scene. Those proposed regions are then given to the zero-shot object detection model, which uses the text prompt to estimate the best matching region. Finally, using the depth data, the object is localized, and the grasping process begins.

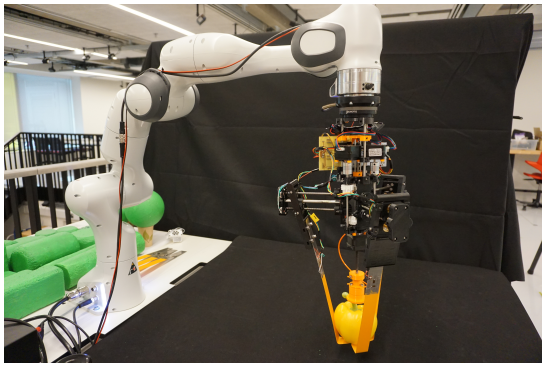

Fig. 1: Object grasping, based on our introduced system, using the manipulator consisting of: a Franka Emika Panda arm, the modified UCL three-finger caging-style gripper, a wrist force/torque sensor, and an RGB-D camera.

We grasp with a caging-inspired gripper [9], designed with three flexible fingers and a movable palm. Caging refers to surrounding and trapping objects, which is an approach tolerant to uncertainties in both object geometry and position. This synergizes with our localization pipeline and few-shot ethos, in particular, by using only in-grasp force sensing data and no camera, as grasping proceeds. We deploy a grasp controller trained with reinforcement learning [10], which uses sensor feedback to control grasping forces, feeling when the grasp is stable and reacting to changes. We evaluate our system using 10 objects from the YCB set [11], testing our pipeline both as isolated elements and as an ensemble to demonstrate effective grasping of unseen objects in clutter, leveraging minimal prior knowledge.

Next, we review the literature, followed by a description of the hardware (gripper and sensors) and software (affordance localization, text-to-image task description, reinforcement learning grasping, and integration developments). Finally, we demonstrate our method in simulation and in the real-world, by comparing grasps of various objects.

II. RELATED WORK

While numerous works have previously explored various aspects of robotic grasping, from teleoperation [12] to autonomous manipulation [13], our method addresses a critical

The authors are with the Department of Computer Science and Department of Mechanical Engineering, University College London, Gower Street, WC1E 6BT, London, UK. {xiang.long.22, luke.beddow, dennis.hadjivelichkov, h.wurdemann, d.kanoulas}@ucl.ac.uk

*equal contribution

This work was supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike), the UCL EPSRC DTP in Fundamental Engineering [EP/T517793/1], and the CDT for Foundational Artificial Intelligence [EP/S021566/1].

Fig. 2: Left to right: the gripper from two viewpoints, the strain gauge finger sensors, and the attached RGB-D camera.

hardware challenge by combining finger compliance, tactile force sensing, closed-loop gripper feedback, and simulated training using reinforcement learning within a single gripper design. This comprehensive integration sets our system apart from previous hardware solutions, such as those discussed in Newbury et al. [3].

In the context of machine learning, where deep reinforcement learning has revolutionized grasping techniques [14], our focus shifts towards safe grasping of delicate objects [15], [16]. Moreover, we embrace the challenge of accomplishing semantic tasks described in natural language, such as "grasp the red mug" [17], [18]. Our method uniquely combines these language-based tasks with our advanced gripper system. Furthermore, we recognize the growing need to reduce reliance on annotated data for grasping learning [19], [20]. Our approach directly addresses these challenges, specifically tailored to the distinctive features of our novel gripper and its multi-modal sensor capabilities.

In summary, our paper introduces a unique combination of contributions by seamlessly integrating hardware innovation, safe grasping, language-based tasks, and reduced annotation requirements, all while achieving high success rates in affordance localization, object identification, and grasping tasks.

III. HARDWARE DESCRIPTION

Our hardware system (visualized in Figs. 1 and 2) is comprised of a 7DoF Franka Emika Panda robotic arm and a caging gripper using in-grasp force sensing [9], [10], which we enhanced further with an RGB-D sensor for the localization pipeline. The gripper aims to grasp objects by surrounding them with three flexible steel fingers. Each of the fingers has a 90° bend at the tip so that they can slip underneath objects. Each finger is equipped with strain gauges, shown in Fig. 2, measuring bending and hence in-grasp forces. The palm is movable and descends to constrain objects, also equipped with force sensing via a penny load cell. There are two main mechanisms in the grasp. Firstly, the caging effect traps objects with the three hooked fingers and palm. Secondly, friction is generated by squeezing the fingers and palm around the object, which is moderated by the flexible finger bending and monitored by the force sensing. A Robotiq force/torque sensor is mounted on the robot wrist above the gripper, which can measure out-of-grasp forces such as the object weight or environment collisions.

The gripper has 3DoF. The three fingers are mechanically coupled, and can be actuated in two ways: prismatically moving in and out or rotating about the base to change their angle. The grasping includes a fourth degree of freedom, as the Panda robot arm moves vertically up and down depending on the grasping controller. All three gripper actuations are achieved by driving leadscrews with stepper motors, ensuring that these motions are non-backdrivable. This improves the caging, as now the object cannot backdrive any motors to relax the geometric constraints.

IV. SOFTWARE DESCRIPTION

Our software system integrates three different parts. The first one (Sec. IV-A) deals with the problem of one-shot object affordances (i.e., meaningful areas on objects) localizing in the scene. The second one (Sec. IV-B) deals with the problem of receiving a text-based task –in our case, a language-based description of the object that needs to be grasped– and localizing an object to be grasped in the scene. The third one (Sec. IV-C) deals with the problem of enabling reinforcement learning to grasp an object using multi-modal sensory data feedback. Connecting the dots (Sec. IV-D) the final integration details are explained. The full methodology is visualized in Fig. 3.

A. One-Shot Object Affordance Localization

Given a novel unseen scene, we are interested in determining which objects can afford to be grasped (or further to be manipulated). There are multiple ways in doing that, e.g., with supervised deep learning [17], [21]. More recently, the research community is moving towards self-supervised [7] or one-shot methods, such as our developed affordance correspondence method, named AffCorrs [8] that we brief in this section and use in the integrated system.

The inputs are: 1) a query RGB image $\mathbf{I}_{query}$ of an object, 2) a binary region mask $M_{query} \in \{0, 1\}$ of some part of the object which is graspable in our case (or manipulable in general), and 3) the RGB image $\mathbf{I}_{target}$ of the target novel scene. Firstly, DINO-ViT features [22] of both images (i.e., query and target) are computed. This allows kNN-like search across semantically similar features as has been shown in [23], [24]. Those features are further cyclically matched

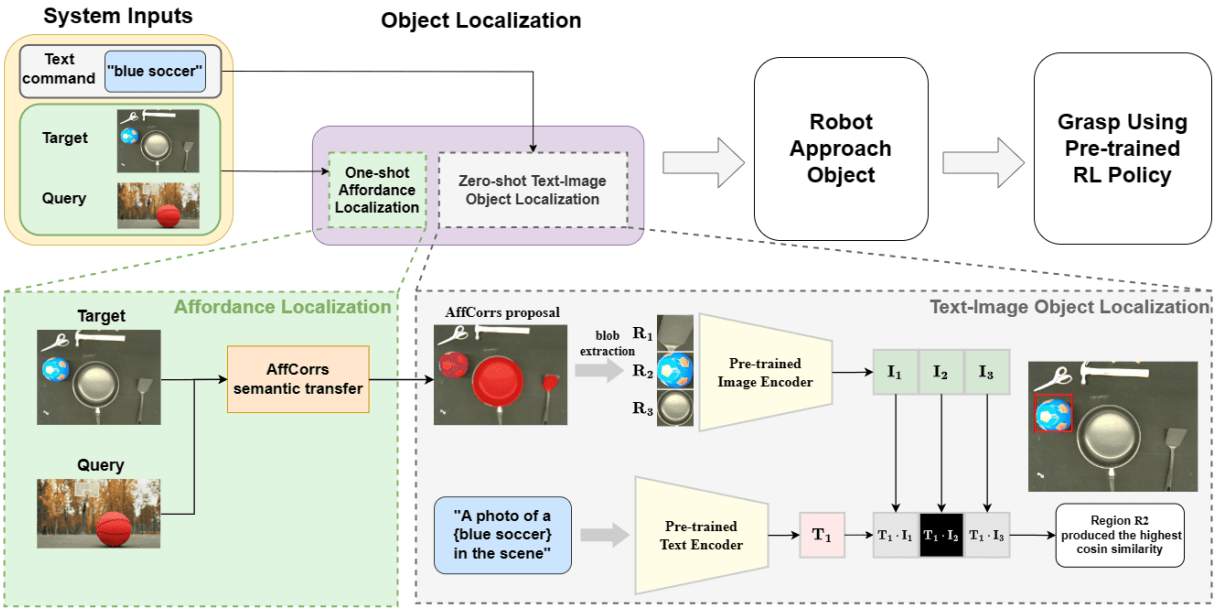

Fig. 3: System overview. AffCorrs (green background) searches regions in the target scene, tha have similar semantic features as the query image, and CLIP (gray background) further corrects it based on a text task description. The robot approaches and grasps the localized object using the caging gripper trained with reinforcement learning.

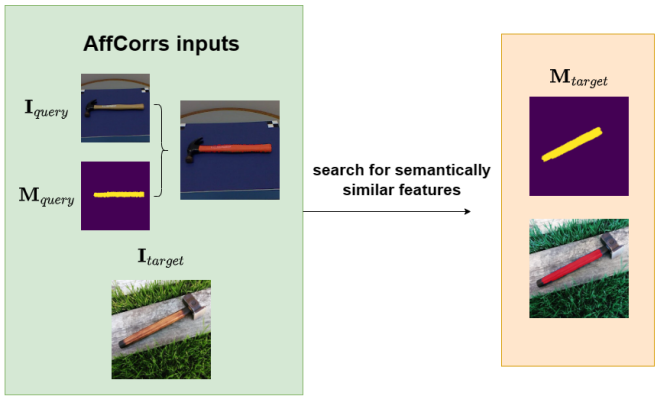

Fig. 4: One-shot grasping affordance correspondence (in red) for a hammer's handle in the scene.

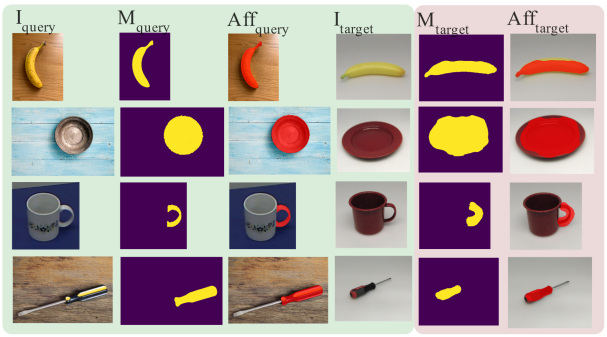

Fig. 5: Four examples showing the one-shot affordance localization for YCB objects.

and a one-to-many region correspondence is computed for regions in the target scene that likely share the similar features as the query region. The corresponding output regions are also represented as binary masks.

The results of one-shot affordance localization is visualized in Fig. 4. This step is important in our work, as given the type of object area we want to manipulate from an example/query, we can find all areas in the scene that have similar affordances. For instance, in Fig. 4, the query image $\mathbf{I}_{query}$ is a hammer, with the binary region mask $M_{query} \in \{0, 1\}$ being its graspable handle. The target image $\mathbf{I}_{target}$ contains a novel hammer of different type and shape. The output of the method is the handle of the novel hammer in the firstly seen target scene. In Fig. 5, there are four output examples we ran on novel YCB objects. Each row visualizes one object sequence of: query RGB image $\mathbf{I}_{query}$, query binary region mask $\mathbf{M}_{query}$, the extracted query affordances $\mathbf{Aff}_{query}$, target novel RGB image of the YCB object $\mathbf{I}_{target}$, target binary region mask $\mathbf{M}_{target}$, and the final affordance correspondence for the novel YCB object $\mathbf{Aff}_{target}$.

B. Zero-Shot Text to Image Task Description

Given the proposed affordance regions as a binary mask in the novel target scene, using the one-shot method in Sec. IV-A above, we treat the detected areas as blobs. Those blobs represent areas that can be grasped. Since there might be more than one graspable area on multiple objects in a scene, we need a way to specify a target object for a grasping task. Thus, an additional pre-trained vision-language model is utilized to evaluate each detected blob, enabling our model to downstream object-level robotic tasks. This concept is known as zero-shot text-to-image object localization. There are three stages, that can be also visualized in the lower right sub-figure in Fig. 3:

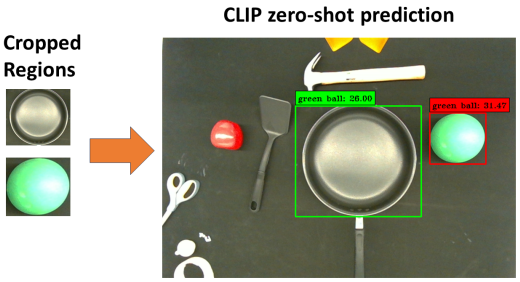

Fig. 6: Zero-shot object detection. The input is two blobs, and the query text is "green ball". The red bounding box in the right sub-figure, denotes the region with the highest confidence score, i.e., the green ball in the novel scene.

1) Blob Extraction: The extraction process takes two inputs that come from the one-shot affordance localization: (i) the RGB image of the target scene $\mathbf{I}_{target}$ and (ii) the affordance binary masks $M_{target} \in \{0, 1\}$. We use the Spaghetti Labeling method [25] to determine the connectivity of blob-like regions in M_{target}. This is a graph and block-based method, which is very efficient, especially when dealing with connected regions in binary images. This method divides an image into small blocks and for each block it creates a graph with nodes (pixels) and edges (connection between pixels) in order to compute and label their connected regions. After all blocks are processed, a post-processing step is performed to make sure the result between blocks are consistent. In this way, all pixels that are of the same binary label are grouped together as blobs. All blob regions are extracted and used for the object localization.

2) Open-Vocabulary Object Detection with Cropped Regions: Having the blobs extracted, we need to provide a task to the system, i.e., in our case, which object should be grasped. For this purpose, we use CLIP [6] to select a region in the novel query image for grasping/manipulation. This is a joint image-text model, trained using contrastive learning. We selected CLIP due to its impressive performance on image-text pairing with novel unseen categories.

We apply this capability to perform zero-shot object detection based on the output AffCorrs affordances and the generated blobs. In particular, given the set of affordances in the novel scene (target image) and the text task, we compute the feature embedding of the blobs and the text, based on the CLIP pre-trained encoding. Then, the cosine similarity of each of the blob-text embedding pair is calculated, and the pair with the highest similarity score is the one that matches our text-based task description. For example, in Fig. 6, our text task input is given as "green ball", and the blobs are those of a gray pan and a green ball; the green ball is detected in the novel scene. The method is also visualized in Fig. 3, lower right part.

3) Object Localization: Last but not least, we localize the object that needs to be grasped/manipulated from the top, by computing the 3D centroid of the selected object region, by combining the original target RGB image $\mathbf{I}_{target}$, region binary mask, predicted by CLIP, and target scene depth $\mathbf{D}_{target}$.

C. Reinforcement Learning-based Grasping

The final stage in our pipeline is grasping. In particular, having detected and localized the objects and the grasping affordance area, the grasping strategy is required to be robust to any accrued positioning errors. We deployed a pre-trained reinforcement learning grasping policy, where a simulated version of the gripper was used for training before direct transfer to the real world. The grasping controller used a deep Q-network to select discrete actions each step, either one of the 3 gripper actuations or vertical motion to lift and lower the gripper.

In keeping with the flexibility of the zero and one-shot ethos of our approach, the grasping does not use vision, but only the sensor data available from the three finger sensors, palm sensor, and wrist F/T sensor. Since forces are continually fed back into the grasping network, the grasp can react to changes via feedback control, to improve the robustness and help compensate for errors. When the controller considers the object to be well grasped, it can lift the object, with a grasp considered successful if the object is lifted to a height of 30mm, not touching the table, and being contacted by all the fingers and the palm.

The RL method [10] was trained on a variety of basic object shapes under the presence of noise to aid generalization. Many food and grocery items which can be approximated to basic shapes [26], well suited to caging. For these objects, caging tolerates uncertainty well, whether this be in object geometry or position, which is why it was chosen to integrate into our approach.

D. System Integration

The system integration is made using ROS. An overview of the integration model is shown in Fig. 3. The three software modules (i.e., affordances, object localization, RL grasping) are integrated to localize an object, an area of the object that can afford grasping, and finally move above the object and start cage grasping it via RL.

V. EXPERIMENTAL VALIDATION

In order to prove that the integration works effectively, we present the experimental validation of the system on a Franka Emika Panda, a Robotiq wrist F/T sensor, and our novel gripper with an attached Intel D405 RGB-D camera.

A. Experimental Protocol

To ensure the robustness of evaluation method, 10 objects are randomly selected from the YCB food items (Fig. 7-left). We trialed each object 5 times, giving 50 total trials for the whole system. Then, we completed extra trials to determine the performance of each component in isolation, 50 trials each. Every trial contained 5 objects, 4 being distractors from the set, which were randomly selected (with a spare) then fixed across batches of 10 trials, one per object. In Fig. 7-right, we show successful object localization during trials,

Fig. 7: **Left:** the query images (top) and the used YCB objects (bottom); **Right:** visual results of object localization via affordance extraction and text task description, for several YCB objects in novel scenes.

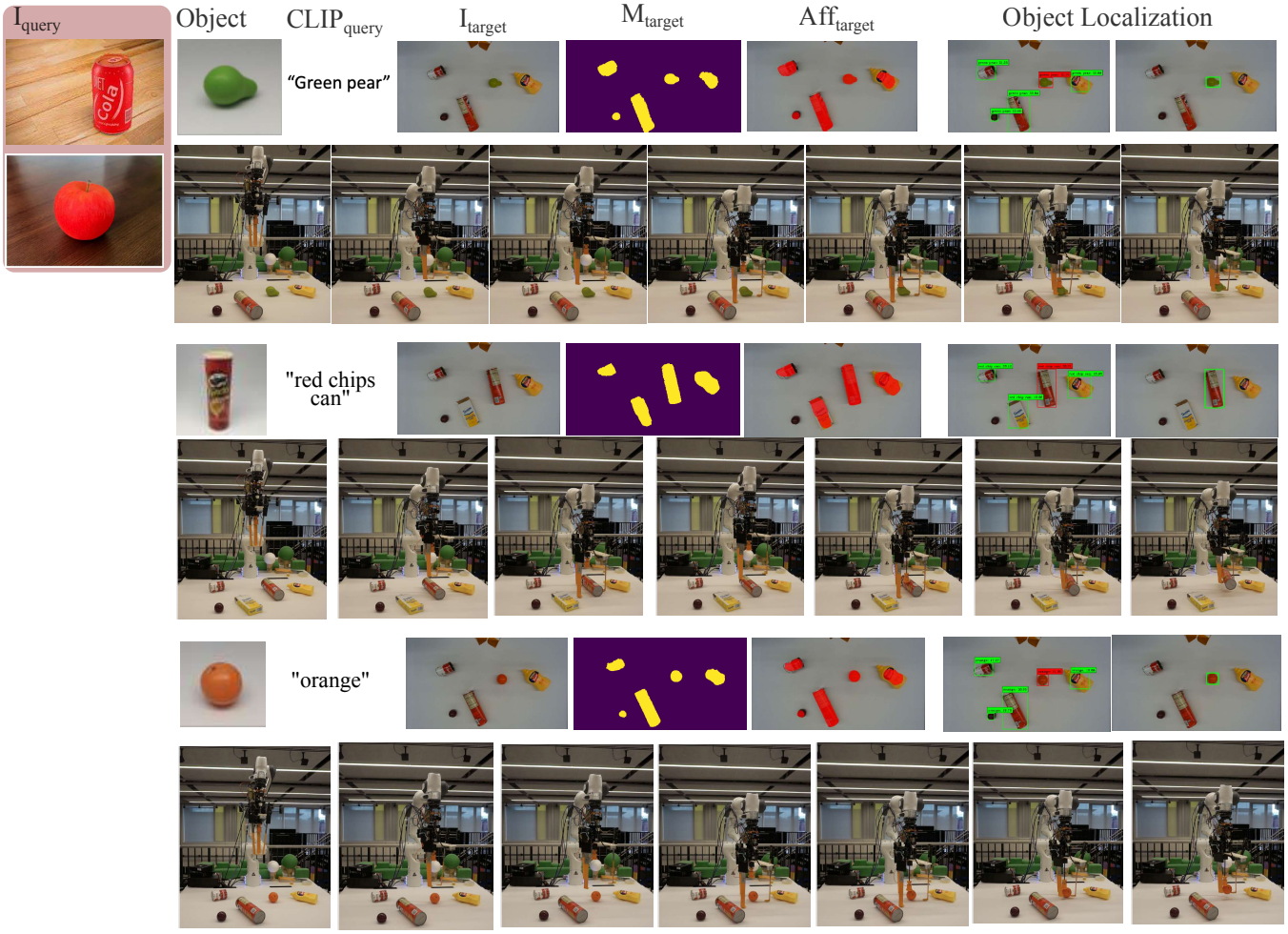

Fig. 8: The grasping results for three YCB objects – pear, can, orange (left to right): the query image (can for the can, and apple for the pear and orange), the text grasping task description, the novel target image, the extracted binary map, the extracted affordances, and the localized object. Below each row, we visualize the sequence of grasping moves by our robot.

while in Fig. 8 we visualize three successful localization and object grasps throughout the full system pipeline.

B. Results

The results for the whole system are shown in Table I. The affordance detection was extremely reliable, achieving 96% success rate, while the CLIP object detection had a 62.5% success rate. When tested in isolation, the grasping approach succeeded 72% of the times, while when all three components ran in sequence, the overall success rate for grasping unseen objects was 48%.

C. Discussion

The integrated system combined several complex components, with an overall 48% grasping success rate. The three individual components of the system had varied reliability,

TABLE I: Integrated system grasping results.

	Component in isolation			All integrated
	Affordance detection	Object identifcation	Grasping	
Success rate / %	96	62.5	72	48

with the 62.5% success rate for the CLIP-based object localization being the key bottleneck. The affordance detection achieved high 96% success, even despite the support images showing only related objects. For example, for all of the fruit objects the support image was an apple. However, AffCorrs tended to pass several possible affordance regions to CLIP, increasing the difficulty of identifying the correct object from the text prompt. We performed little to no prompt engineering, however, it is likely that reliability would be improved with more targeted prompts. The grasping had a 72% success rate, grasping every object at least once in the 5 trials. The moderate and larger objects were well grasped, with worst performance on the lemon and Strawberry Jello box, two of the smallest objects.

VI. CONCLUSIONS AND FUTURE WORK

In conclusion, we presented an integrated system combining one-shot affordance localization, zero-shot object identification, and reinforcement learning grasping. The system took as input an image of a related object to the target, as well as up to three words of text describing the target object. The entire system focused on few-shot methods and flexibility, with grasping proceeding without a camera, instead caging using in-grasp force sensing only. The overall system success rate on 10 YCB objects was 48%, with good affordance detection of 96% and grasping of 72%, but object detection being the primary limiting factor with 62.5% success rate. Future work will focus on improving the localization system by reducing the number of affordance regions passed to the object localization, and improving the grasping itself with improved fingertip designs and gradient based reinforcement learning methods.

REFERENCES

[1] P. Balatti, D. Kanoulas, N. G. Tsagarakis, and A. Ajoudani, "Towards Robot Interaction Autonomy: Explore, Identify, and Interact," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[2] J. Liu *et al.*, "Garbage Collection and Sorting with a Mobile Manipulator using Deep Learning and Whole-Body Control," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2021.

[3] R. Newbury *et al.*, "Deep Learning Approaches to Grasp Synthesis: A Review," *IEEE Transactions on Robotics*, pp. 1–22, 2023.

[4] D. Kanoulas, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, "Center-of-Mass-Based Grasp Pose Adaptation Using 3D Range and Force/Torque Sensing," *International Journal of Humanoid Robotics*, 2018.

[5] Z. Xie, X. Liang, and C. Roberto, "Learning-based Robotic Grasping: A Review," *Frontiers in Robotics and AI*, vol. 10, 2023.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *38th International Conference on Machine Learning (ICML)*, 2021, pp. 8748–8763.

[7] D. Hadjivelichkov and D. Kanoulas, "Fully Self-Supervised Class Awareness in Dense Object Descriptors," in *Conference on Robot Learning*, 2022, pp. 1522–1531.

[8] D. Hadjivelichkov, S. Zwane, M. Deisenroth, L. Agapito, and D. Kanoulas, "One-Shot Transfer of Affordance Regions? AffCorrs!" in *Conference on Robot Learning*, 2023, pp. 550–560.

[9] L. Beddow, H. Wurdemann, and D. Kanoulas, "A Caging Inspired Gripper using Flexible Fingers and a Movable Palm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7195–7200.

[10] ——, "Grasping by Touch: Combining Reinforcement Learning with Compliant Fingers," 2024.

[11] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[12] E.-J. Rolley-Parnell *et al.*, "Bi-Manual Articulated Robot Teleoperation using an External RGB-D Range Sensor," in *15th International Conference on Control, Automation, Robotics and Vision*, 2018.

[13] Kanoulas, Dimitrios and Lee, Jinoh and Caldwell, Darwin G. and Tsagarakis, Nikos G., "Visual Grasp Affordance Localization in Point Clouds using Curved Contact Patches," *International Journal of Humanoid Robotics (IJHR)*, 2017.

[14] H. Zhang, J. Tang, S. Sun, and X. Lan, "Robotic Grasping from Classical to Modern: A Survey," *arXiv:2202.03631*, 2022.

[15] V. Gabler, G. Huber, and D. Wollherr, "A Force-Sensitive Grasping Controller Using Tactile Gripper Fingers and an Industrial Position-Controlled Robot," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 770–776.

[16] C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, "Task-Oriented Grasp Prediction with Visual-Language Inputs," *arXiv preprint arXiv:2302.14355*, 2023.

[17] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with Convolutional Neural Networks," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2016.

[18] K. Xu, S. Zhao, Z. Zhou, Z. Li, H. Pi, Y. Zhu, Y. Wang, and R. Xiong, "A Joint Modeling of Vision-Language-Action for Target-oriented Grasping in Clutter," *arXiv preprint arXiv:2302.12610*, 2023.

[19] M. A. Lee *et al.*, "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," in *Int. Conference on Robotics and Automation*, 2019, pp. 8943–8950.

[20] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning Task-Oriented Grasping for Tool Manipulation from Simulated Self-Supervision," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.

[21] A. Nguyen *et al.*, "Object-based Affordances Detection with Convolutional Neural Networks and Dense Conditional Random Fields," in *IEEE/RSJ IROS*, 2017, pp. 5908–5915.

[22] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in *International Conference on Computer Vision*, 2021.

[23] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep ViT Features as Dense Visual Descriptors," *arXiv preprint arXiv:2112.05814*, 2021.

[24] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in *International Conference on Computer Vision*, 2021.

[25] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *IEEE T. on Image Processing*, vol. 29, pp. 1999–2012, 2020.

[26] W. Friedl and M. A. Roa, "CLASH—A compliant sensorized hand for handling delicate objects," *Frontiers in Robotics and AI*, vol. 6, pp. 1–15, 2020.