Manufacturing Letters 42 (2024) 16–20

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

Letters

Zero-shot autonomous robot manipulation via natural language

Changheon Han [a], Jiho Lee [a], Hojun Lee [a], Yuseop Sim [a], Jurim Jeon [b], Martin Byung-Guk Jun [a,*]

[a] *School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette 47907-2088, USA*
[b] *School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjung-gu, Busan, 46241, South Korea*

ARTICLE INFO

Article history:
Received 13 April 2024
Received in revised form 27 August 2024
Accepted 2 September 2024
Available online 28 September 2024

Keywords:
Autonomous manufacturing
Human-machine interaction
Zero-shot robot manipulation
Large Language Model (LLM)
Natural language

ABSTRACT

Smart manufacturing revolutionizes advanced automation by leveraging deep learning and robotics technology. Nevertheless, deep learning implementation in manufacturing faces several challenges in preparing datasets, training a model, and adapting to various applications. Furthermore, programming languages are required to develop deep learning models and communicate with robots, which is not straightforward to quickly and precisely reflect human intentions in manufacturing processes. To alleviate the issue, this study proposes an autonomous zero-shot robot manipulation framework via natural language. Specifically, the Segmentation Anything Model (SAM), a promptable image segmentation model, first segments objects in images captured by an RGB-D camera, which are passed to Generative Pre-trained Transformers (GPTs), a Large Language Model (LLM), to identify and localize a designated object based on human commands given in natural language. Afterwards, the Robot Operating System (ROS) manipulates a robot to a localized position. With no additional programming or specialized skills in object localization and robot manipulation, the framework demonstrates an autonomous robot automation approach through intuitive communication between humans and machines through natural language.

1. Introduction

The advancement of Information and Communication Technology (ICT) has profoundly influenced the manufacturing domain, paving the way for the development of smart manufacturing [1]. Within ICT, deep learning has become paramount in smart manufacturing thanks to its ability to automate decision-making, streamline operations, and enhance productivity [2]. However, preparing datasets and building models require substantial resources [3]. While our previous investigations addressed this by automating the generation and annotation of datasets through multi-stage architectures incorporating computer vision techniques [4,5,6], such applications remain confined to narrow objectives and impede seamless interaction with humans and other entities, relying on programming languages [7]. This indirect interaction hinders the proper reflection of human intention and context in manufacturing scenarios.

These challenges have underscored the necessity for more universal deep learning approaches, which leads this investigation to implement Large Language Models (LLMs) as they can process underlying intention and context in human language. Generative Pre-trained Transformers (GPTs), a type of LLM, extend this capability, analyzing diverse media such as images and audio. Despite their proven versatility and efficiency in everyday applications, their utilization within the manufacturing sector remains nascent due to integration complexities and a lack of domain-specific case studies [8].

Thus, this study proposes a framework for autonomous zero-shot robot operation. Utilizing the Segment Anything Model (SAM) [9], a promptable image segmentation model capable of effectively segmenting images from a manufacturing environment [10], and GPTs, the framework identifies and localizes an object based on commands expressed in natural language. The Robot Operating System (ROS) [11] was utilized to manipulate a robot according to the location of the identified object. This framework eliminates the need to train any deep learning model and showcases the ability to comprehend human intention and the context of a manipulation scenario through natural language.

2. Methodology

The proposed framework aims to autonomously localize an object specified by a user in natural language and manipulate a

[*] Corresponding author.
E-mail address: mbgjun@purdue.edu (M.B.-G. Jun).

https://doi.org/10.1016/j.mfglet.2024.09.067

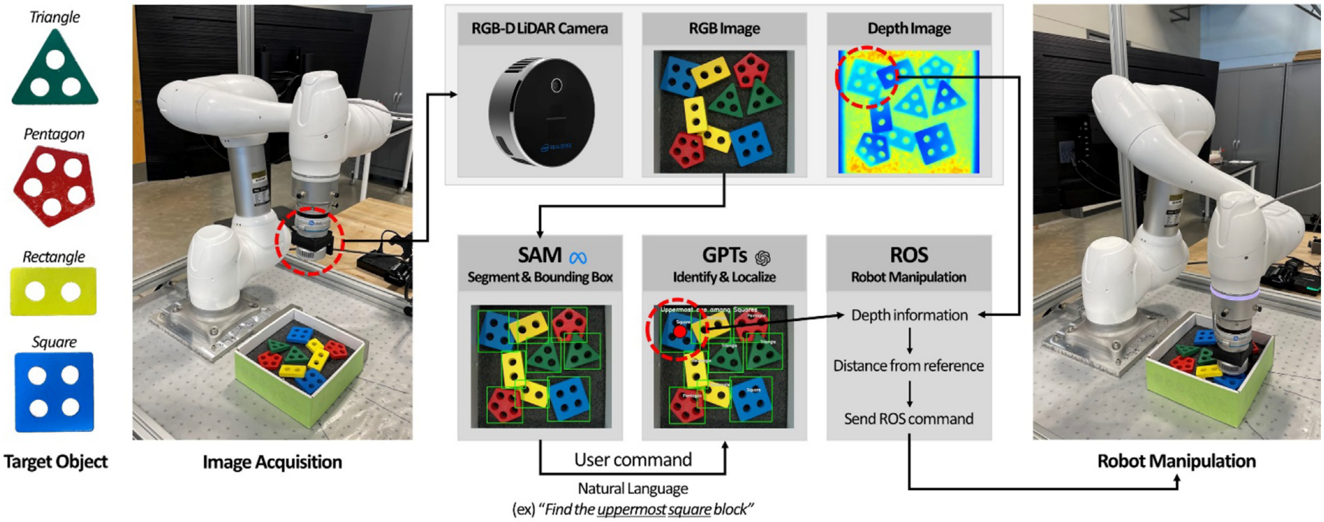


Fig. 1. Overview of the Proposed Framework.

robot to the identified location without training additional deep learning models. Fig. 1 illustrates the overview of the framework. The setup began with spreading a set of triangle, square, rectangle, and pentagon blocks inside a box. An RGB-D camera (Intel RealSense L515) attached to an end effector of a robot vertically captured RGB and depth images of these blocks from a distance of 350 mm above the box with a resolution of 640x480. This study applied the ViT-H SAM, the largest pre-trained image segmentation model among the SAMs, to segment an RGB image. The segmented images were kept with sizes ranging from 80x80 to 180x180 to minimize data contamination while a corresponding JSON file was created to record their coordinates in the RGB image.

In the subsequent step, the GPT-4–1106-vision-preview model (GPT-4-vision) identified the object type in each segmented image, and another JSON file recorded their types and coordinates. The GPT-4–0125-preview model (GPT-4) then processed a user's command which included information about an object type and its relative location in natural language and returned the coordinates of the object matched to the requested object type and relative position. Based on the three-dimensional locations in the depth image corresponding to the returned coordinate, robot coordinates were calculated. Then, the ROS issued a command to manipulate a robot toward the designated object. This investigation utilized a workstation operating on Ubuntu 20.04.5 LTS with an Intel i7-11700 K CPU, 64 GB RAM, and Nvidia RTX A5000 GPU for computation tasks and a Doosan Robotics m0609 collaborative robot for manipulation purposes.

3. Results & Discussion

This study evaluates the performance of the proposed framework based on its accuracy in estimating the type and location of an object according to a user's prompt. Table 1 describes the prompts used for each GPT model to examine their impacts on the framework performance by type of information. 'object type' is a variable in the Identification Prompts (IPs), which identify the type of an object by the GPT-4–1106-vision-preview model (GPT-4-vision). The Identification Prompt #1 (IP 1) contains only the types of objects in an image, while subsequent prompts include additional details such as the shape definition, the number of holes, or the color of each object. The Localization Prompts (LPs), which localize a target object by the GPT-4–0125-preview model (GPT-4), include variables of 'object type' and 'position' (leftmost, rightmost, uppermost, and bottommost). The localization prompt #1 (LP 1) has the content of a JSON input file only, whereas LP 2 specifies data in a JSON file for comparison.

Fig. 2 presents the prediction results obtained using IPs and LPs. This study evaluated the performance of IPs across 82 cases involving 7 images of scattered objects. While including color information (IP 5–8) enhanced object identification performance, the IPs containing details without color (IP 2–4) made accuracy and performed even worse than IP 1, the prompt without additional description.

For IP 2, the model prioritized shape definitions to determine object types. Consequently, when the sides of an object were hidden or indistinct, the model using IP 2 struggled to define the object type. This limitation was particularly evident when objects were overlapped by another object of the same color, resulting in incorrect identification. For instance, Fig. 3 (b) depicts that the model labeled the rightmost and uppermost rectangles as unknowns. However, the model correctly identified the bottom-right pentagon covered by other objects with different colors.

In contrast, the model using IP3 demonstrated improved performance by identifying objects based on the number of holes along with the shape definitions. It enabled the model to recognize object types even when overlapped by objects of the same color since the identification relied on information irrelevant to shapes. Nevertheless, the model using IP3 still faced challenges in identifying objects with incomplete information about holes. For example, one hole of the pentagon at the center in Fig. 3 (c) was hidden, leading the model to label the pentagon as unknown. In summary, the model with the prompts based on information about shapes and holes faced challenges in identifying objects with incomplete information and showed accuracies ranging from 64.63 to 80.49 %.

On the other hand, the model incorporating color information achieved accuracies ranging from 92.7 % to 100.0 %, as colors proved to be distinctive features for identifying vague or occluded objects. Notably, IP 8 correctly identified the combined segment in Fig. 3 (d) as both a triangle and a rectangle, while IP 1, IP 2, and IP 3 failed to do so. By integrating geometric and non-geometric information, the GPT-4-vision model could thoroughly identify objects. Consequently, it suggests utilizing geometric and non-geometric information in a prompt. Setting priorities between criteria would be an option.

Subsequently, this study evaluated the performance of the Localization Prompts (LPs) on the objects accurately identified by IP 8. Fig. 3 (e) and (f) illustrate the estimated location of the designated object with a black-outlined red circle. LP 1, which failed to completely localize the designated object, achieved an accuracy of 92.31 %. In contrast, LP 2, which specifies which data should be used for position comparison, reached 100 % accuracy in localizing the object. This improvement highlights the importance of specifying the exact data type for analysis, thereby enhancing the text analysis capabilities of the GPT model. Given that GPTs have multiple analyzers, this result underscores the necessity of defining their specific roles within a task to achieve optimal performance.

Finally, the physical distance from the reference point (the center of the captured image) to the estimated location of the designated object, as determined by LP2 in the x and y directions, was calculated using depth data. Then, the robot was manipulated to position itself above the object through the ROS based on the computed distances (Fig. 1).

4. Conclusion

The proposed zero-shot framework demonstrated the capability of targeting objects via natural language with 100 % accuracy. Instead of developing an object detection model, this approach specified and localized an object based on the description written in natural language. With no additional programming, dataset preparation, and deep learning model development, it could be possible to identify and locate even a new object with a natural language-based description, allowing workers without programming skills would manipulate a robot. Thus, this framework could aid autonomy in manufacturing scenarios that cannot afford to develop deep learning models while frequent changes in target objects are necessary. However, the practical application of LLMs in real-world industrial settings requires further investigation since well-defined prompts are essential for LLMs that handle natural language encompassing a variety of meanings and contexts and serve multiple roles in universal tasks. Besides, most objects in manufacturing lack textures and distinguishable colors, posing challenges for LLM-based identification. The decision-making process of LLMs is akin to a black box, making it hard to analyze.

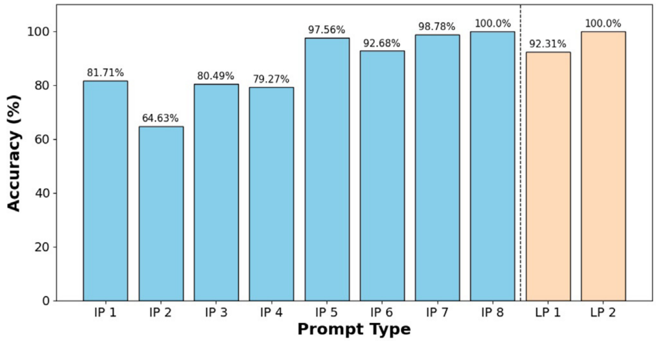


Fig. 2. Prediction results by prompts.

Table 1
Prompts on LLMs.

(1) GPT-4–1106-vision-preview model – Object Identification Prompts (IPs)

Identification Prompt #1 (IP 1: Non-additional information)
There are four types of objects in an image
Triangle, Pentagon, Rectangle, and Square.
If you identify an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

Identification Prompt #2 (IP 2: Definition)
There are four types of objects in an image.
Here is the information about the objects:
Triangle: 3 corners, 3 sides
Pentagon: 5 corners, 5 sides
Rectangle: 4 corners, 4 same-angle sides
Square: 4 same-angle corners, 4 same-angle sides
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

Identification Prompt #3 (IP 3: # Holes)
There are four types of objects in an image.
The blocks have circular holes in them.
Here is the information about the objects:
Triangle: 3 holes
Pentagon: 5 holes
Rectangle: 4 holes
Square: 4 holes
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

Identification Prompt #4 (IP 4: Definition, # Holes)
There are four types of objects in an image.
The blocks have circular holes in them.
Here are the information of the objects:
Triangle: 3 holes, 3 corners, 3 sides
Pentagon: 5 holes, 5 corners, 5 sides
Rectangle: 2 holes, 4 corners, 4 same-angle sides
Square: 4 holes, 4 same-angle corners, 4 same-angle sides
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

Identification Prompt #5 (IP 5: Color)
There are four types of objects in an image.
Here are the information of the objects:
Triangle: Green / Pentagon: Red
Rectangle: Yellow / Square: Blue
If you identify an *object type* is in the following image, return the object type.
If you cannot identify the object, return 'unknown'.

Identification Prompt #6 (IP 6: Color, Definition)
There are four types of objects in an image.
Here is the information about the objects:
Triangle: Green, 3 corners, 3 sides
Pentagon: Red, 5 corners, 5 sides
Rectangle: Yellow, 4 corners, 4 same-angle sides
Square: Blue, 4 same-angle corners, 4 same-angle sides
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

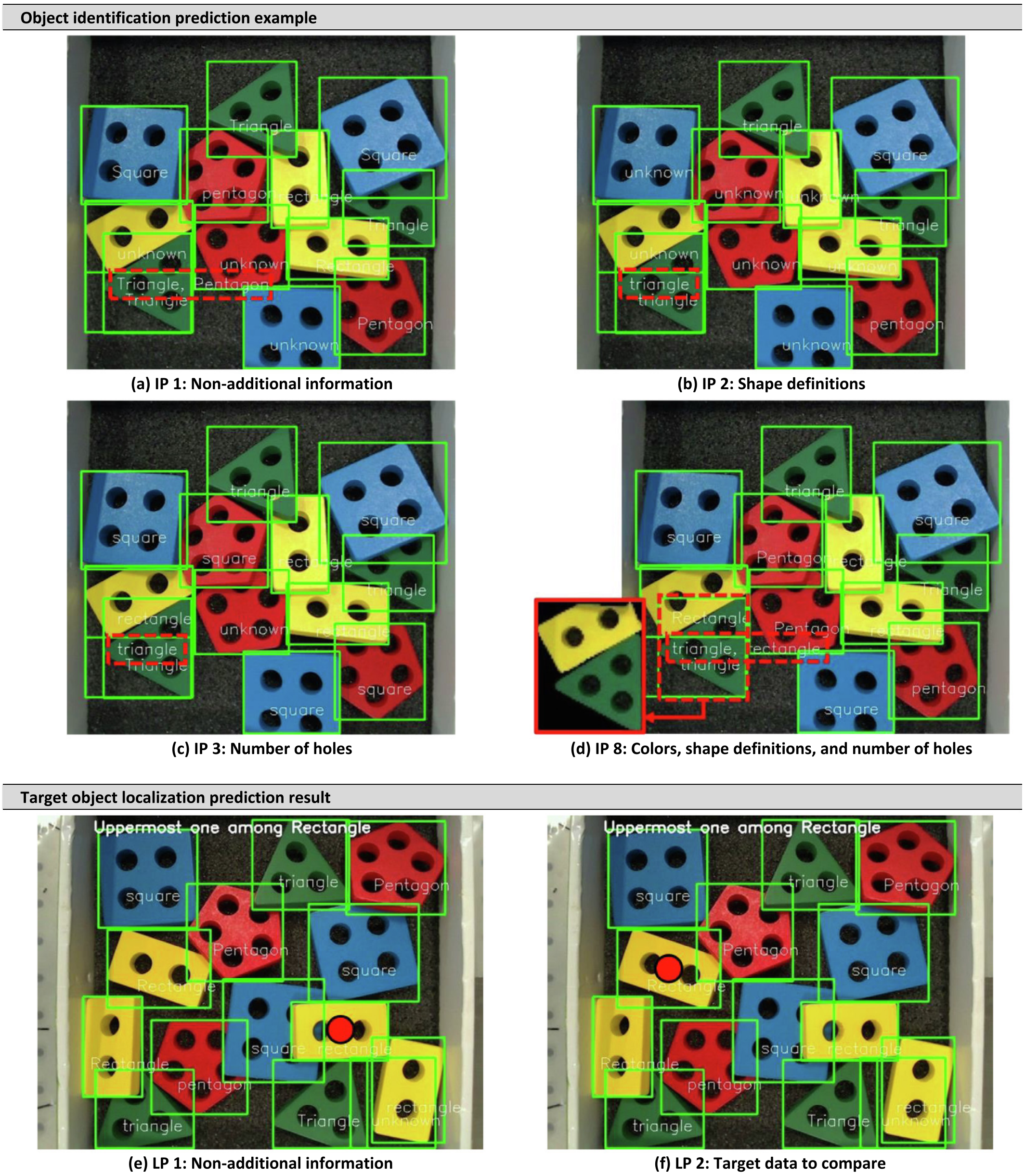
Identification Prompt #7 (IP 7: Color, # Holes)
There are four types of objects in an image.
The blocks have circular holes in them.
Here is the information about the objects:
Triangle: Green, 3 holes / Pentagon: Red, 5 holes
Rectangle: Yellow, 4 holes / Square: Blue, 4 holes
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

Identification Prompt #8 (IP 8: Color, Definition, # Holes)
There are four types of objects in an image.
The blocks have circular holes in them.
Here are the information of the objects:
Triangle: Green, 3 holes, 3 corners, 3 sides
Pentagon: Red, 5 holes, 5 corners, 5 sides
Rectangle: Yellow, 2 holes, 4 corners, 4 same-angle sides
Square: Blue. 4 holes, 4 same-angle corners, 4 same-angle sides
If you identify that an *object type* is in the following image, return the object type.
If you cannot identify the object, return unknown.

(2) GPT-4–0125-preview model – Target object localization prompts (LPs)

Localization Prompt #1 (LP 1: No additional information)
This JSON file includes a filename, an object type, and x and y-centers.
Find an object at the *position* one among the *object type* in the JSON file.
Then, return the x and y-centers as the format: (x,y) without details.

Localization Prompt #2 (LP 2: Define data to compare)
This JSON file includes a filename, an object type, and x and y-centers.
Compare the x-center and y-center of the object types in the JSON file.
Find an object at the *position* one among the *object type* in the JSON file.
Then, return the x and y-centers as the format: (x,y) without details.



(a) IP 1: Non-additional information

(b) IP 2: Shape definitions

(c) IP 3: Number of holes

(d) IP 8: Colors, shape definitions, and number of holes

(e) LP 1: Non-additional information

(f) LP 2: Target data to compare

Fig. 3. Examples of prediction results by prompts.

Future research will focus on developing guidelines for effective LLM prompts in manufacturing, integrating other pre-trained models, fine-tuning LLMs, and exploring the use of diverse data sources, such as sound. Investigating methods to analyze the decision-making process of LLMs will also be a key area of exploration.

CRediT authorship contribution statement

Changheon Han: Writing – original draft, Methodology, Investigation, Conceptualization. **Jiho Lee:** Writing – review & editing, Visualization, Supervision, Formal analysis, Conceptualization. **Hojun Lee:** Writing – review & editing, Validation, Methodology, Formal analysis, Data curation. **Yuseop Sim:** Writing – review & editing, Validation, Software, Investigation, Formal analysis. **Jurim Jeon:** Methodology, Investigation, Formal analysis, Data curation. **Martin Byung-Guk Jun:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Zheng P et al. Smart manufacturing systems for Industry 4.0: conceptual framework, scenarios, and future perspectives. Front Mech Eng 2018;13(2):137–50. https://doi.org/10.1007/s11465-018-0499-5.

[2] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. J Manuf Syst 2018;48:144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.

[3] Kusiak A. Smart manufacturing must embrace big data. Nature 2017;544(7648):23–5. https://doi.org/10.1038/544023a.

[4] Park J, Han C, Jun MBG, Yun H. Autonomous robotic bin picking platform generated from human demonstration and YOLOv5. J Manuf Sci Eng 2023;145 (121006). https://doi.org/10.1115/1.4063107.

[5] Han C, Lee J, Jun MBG, Lee SW, Yun H. Visual coating inspection framework via self-labeling and multi-stage deep learning strategies. J Intell Manuf 2024. https://doi.org/10.1007/s10845-024-02372-9.

[6] C. Han *et al.*, ''Hybrid Semiconductor Wafer Inspection Framework via Autonomous Data Annotation,'' *Journal of Manufacturing Science and Engineering*, pp. 1–34, Apr. 2024, doi: 10.1115/1.4065276.

[7] Zhang T, Gao C, Ma L, Lyu M, Kim M. An Empirical Study of Common Challenges in Developing Deep Learning Applications. In: in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. p. 104–15. https://doi.org/10.1109/ISSRE.2019.00020.

[8] L. Makatura *et al.*, ''How Can Large Language Models Help Humans in Design and Manufacturing?,'' Jul. 25, 2023, *arXiv*: arXiv:2307.14377. doi: 10.48550/arXiv.2307.14377.

[9] A. Kirillov *et al.*, ''Segment Anything,'' Apr. 05, 2023, *arXiv*: arXiv:2304.02643. doi: 10.48550/arXiv.2304.02643.

[10] K. Moenck *et al.*, ''Industrial Segment Anything – a Case Study in Aircraft Manufacturing, Intralogistics, Maintenance, Repair, and Overhaul,'' Jul. 24, 2023, *arXiv*: arXiv:2307.12674. doi: 10.48550/arXiv.2307.12674.

[11] M. Quigley *et al.*, ''ROS: an open-source Robot Operating System,'' in *ICRA workshop on open source software*, Kobe, Japan, 2009, p. 5.