Received: September 20, 2023 | Revised: December 12, 2023 | Accepted: December 14, 2023

JOURNAL OF APPLIED RESEARCH ON SCIENCE AND TECHNOLOGY
JARST
INSTITUTE OF RESEARCH AND DEVELOPMENT
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ISSN ONLINE: 2773-9473
https://ph01.tci-thaijo.org/index.php/
rmutt-journal

# Optimizing pick-place operations: Leveraging k-means for visual object localization and decision-making in collaborative robots

**Naphat Yenjai and Nattasit Dancholvichit***

Department of Mechatronics Engineering, Faculty of Engineering, Rajamangala University of Technology Rattanakosin, Nakhon Pathom 73170, THAILAND

*Corresponding author: nattasit.dan@rmutr.ac.th

## ABSTRACT

This article presents an approach to object localization algorithms for pick-place operations in collaborative robots by utilizing conventional color segmentation in computer vision and k-means clustering. Adding the k-means clustering algorithm complements the color segmentation by distinguishing and grouping the sections of similar pixels; hence, object localization is more accurate. The order of pick-place operations of each cluster acquired from the proposed algorithm is prioritized based on $L^2$ norm. Integrating the proposed framework provides a well-structured depiction of the localized objects, which is fundamental for successful pick-place operations. The TCP/IP communication framework via socket communication is established to facilitate data transmission between the robot and the host computer. The objective is to ensure that the robot's end effector performs as directed by the host computer by obtaining information on the pick-and-place operation, including the localized coordinates, dimensions, the order of operations, and the pose of the objects of interest to the robot. In this experiment, a cobot arm is employed to autonomously pick and place objects with different shapes and colors in a workspace filled with diverse objects, requiring the robot to choose the closest objects to operate based on the data from the host computer. Our results demonstrate the effectiveness of this integration, showcasing the enhanced adaptability and efficiency of pick-place operations in collaborative robots. This study indicates 98% accuracy in pick-and-place operations with an average latency of 0.52 ± 0.1 s, indicating an improvement compared to the traditional algorithm without k-means clustering, which achieves an accuracy of 88%. Additional studies reveal that when incorporating pose estimation into the pick-place operations, the proposed algorithm's accuracy is 94%. The demonstration highlights the potential of leveraging machine learning algorithms and computer vision from the camera to perform flexible pick-place operations via socket communication.

*Keywords:* Cobots, Pick-place operations, K-means clustering, Object localization

## INTRODUCTION

The emergence of Industry 4.0 and the intelligent factory concept has introduced a range of significant technological advancements with the potential to facilitate the development of intelligent products and services [1]. Creating a versatile and adaptable assembly system necessitates integrating various enabling resources and technologies. Collaborative robots, or cobots, are emerging as a technology that offers enhanced flexibility and quick adaptation in assembly processes. Unlike traditional industrial robots, cobots can work alongside individuals without fencing or enclosure [2]. Human-robot collaboration (HRC) involves humans and cobots working together in the same workspace to execute manufacturing processes, leveraging the strengths of both for task completion [3]. Designing a collaborative human-robot workplace poses challenges, requiring adherence to specific design guidelines [4]. These processes can include pick-and-place tasks [5], assembly [6], screwing [7], or inspection [8]. Although cobots demonstrate exceptional performance in collaborating on complex tasks with humans, their reliance on expensive industrial-grade cameras presents limitations in flexibility and adaptability to diverse industrial applications. The cost-intensive nature of implementing and reprogramming robots and their limited ability to effectively navigate dynamically changing environments pose a significant obstacle





DOI: 10.60101/jarst.2023.254153

for small companies seeking to integrate cobots into production lines.

By harnessing the potential of computer vision coupled with machine learning, cobots can potentially undertake more intricate tasks in collaboration with humans, thereby increasing the overall complexity of operations. Typically, engineers input basic commands or position controls in the native language of the robot to enhance ease of use, simplifying both operation and configuration [9, 10]. However, this also constrains their versatility, narrowing down the range of tasks they can effectively perform as these high-level languages are limited to a small number of applications [11, 12]. Compared to industrial-grade cameras, cost-effective cameras in the current market are widely adopted to assist with machine vision tasks. This solution enables cobots to effectively collaborate with humans, performing product quality control and component assembly inspection tasks. Cobots can perceive their surroundings, perform tasks, and make informed decisions through network communication, achieving comparable performance to high-end devices. By setting up a system where cameras are linked to a host computer and provide data to the cobot through TCP/IP (Transmission Control Protocol/ Internet Protocol) [13], the computer can process the visual data captured by the webcams and transmit it to the cobot. With this cost-effective implementation, the cobot can effectively collaborate with the human operator and perform flexible tasks based on computer vision analysis.

Pick-place operations play a fundamental role in various robotic applications. While these operations have become well-established in structured scenarios, challenges arise when dealing with parts of high variability or in less structured environments, especially for HRC. In such cases, pick-place operations are limited to mostly laboratory settings and have not been widely adopted in the market due to inefficiency, lack of robustness, and limited flexibility in existing manipulation and perception technologies [14]. Numerous studies have addressed these challenges to enhance HRC by enhancing object recognition, localization, and pick-and-place operations. In [15], computer vision and image transformation are used to determine the location of the objects to perform pick-place operations. Support Vector Machines (SVM) have been employed to classify successful and failed scenarios [16], while the Point Distribution Model (PDM) has been used to compute generalized success models [17]. In recent studies, most of the work focuses on machine learning algorithms developed where multiple stages of object classification are implemented, including deep learning and point cloud processing [18]. Another recent study explores object detection and recognition in the context of a pick-and-place robot, focusing on edge detection and feature extraction utilizing an Artificial Neural Network (ANN) [19]. Multiple Reinforcement Learning (RL) techniques [20] are also explored to perform task-specific operations, such as pick-place operations without direct programming. These studies provide algorithms that yield high accuracy, providing the computational power is high and sufficient to operate such approaches. Meanwhile, the k-means algorithm has found application in various machine vision scenarios used with cobots, such as in pick-and-place and sorting operations, notably in handling cherry tomatoes. The algorithm uses a color patch-based visual tracking algorithm to precisely detect and pick ripe tomatoes [21]. Another relevant research area pertains to hand gesture recognition as an input command for the Bioroid Premium Robot, with studies exploring the use of k-means clustering and SVM techniques [22].

While computer vision has been extensively utilized in various applications, its reliance on complex image processing algorithms and manual tuning parameters can lead to challenges in achieving high accuracy and robustness. In specific scenarios, sophisticated approaches require substantial computational resources. By incorporating simpler machine learning algorithms into the vision system, it becomes feasible to optimize the framework. This involves decreasing reliance on intricate vision algorithms and harnessing the power of machine learning to enhance accuracy in object recognition, localization, and manipulation tasks. In this study, we propose an algorithm that enhances object localization accuracy by leveraging both machine vision and k-means clustering. This study employs an eye-on-hand approach, focusing on retrieving and performing pick-place operations on a specific object chosen by the user and placing it in a designated target location. A camera is used for capturing images, and the algorithm utilizes computer vision and k-means clustering for improved performance. The host computer employs OpenCV for color segmentation, and the k-means algorithm identifies pick-and-place locations in order. The algorithm also determines the location of the desired object from others and prioritizes the sequence of retrieval based on its $L^2$ norm. The implementation involves the targeted object's decision algorithm and pick-place operations, with predefined trajectories stored on the host computer. The coordinate and pose information is then transferred to the cobot to



DOI: 10.60101/jarst.2023.254153

manipulate the designated object precisely via socket communication.

This article first introduces the experimental setup, followed by a methodology section that describes the algorithms and communication techniques employed in the study. The results and discussion section are presented next. Finally, the conclusions are given.

## MATERIALS AND METHODS

*Experimental setup*

In this experiment, a cobot arm is used to pick and place objects of different shapes and colors. The workspace contains multiple objects of varying shapes and sizes, and the robot must identify the objects of interest and prioritize them for picking and placing. The machine must be able to identify the objects of interest and prioritize the order of pick-place operations such that the closest objects are selected first. In this study, Gofa IRB 15000 Cobot, manufactured by ABB equipped with an OAK-D LITE camera [23], was used to capture images of objects in eye-on-hand configuration, as shown in Figure 1. The IRB 15000 is capable of a payload of 5 kg, a reach of 1.62 m, a speed of up to 2.2 m/s, and a resolution of 0.02 mm.

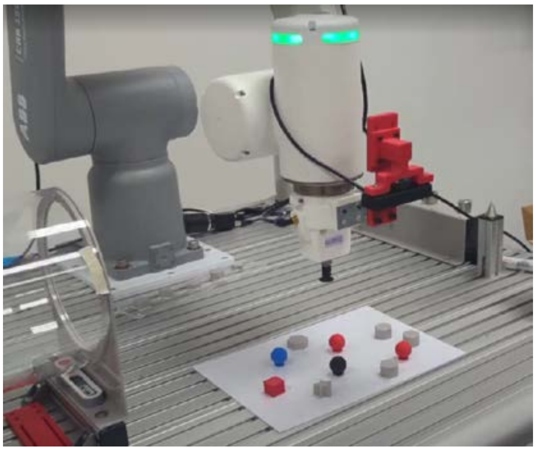


**Figure 1** Cobot is equipped with an eye-on-hand camera to be used in this study of pick-place operations.

Once the experimental session starts, the algorithm, which incorporates computer vision techniques and the k-means clustering algorithm for object localization, is executed. The computer vision component analyzes the visual data captured by the camera, extracting relevant information about the objects present in the scene. This information includes object shapes, positions, and other features necessary for object recognition and localization. The number of pick-place operations the cobot performs throughout the experimental session is counted and recorded. This measurement serves as a metric to evaluate the efficiency and effectiveness of the algorithm and the system's overall performance. It helps assess the ability of the algorithm to locate objects accurately, determine appropriate pick-and-place sequences, and successfully execute the manipulations.

*Methodology*

The proposed methodology for this study is divided into three main parts, as shown in Figure 2. First, the PC host is where the computation of the picking location takes place. It consists of the preprocessing, the segmentation algorithm, and the k-means algorithm, respectively. Then, the host sends command data to the collaborative robot software via socket communication, which ensures efficient object manipulation and interaction. Finally, the cobot receives information about the objects of interest and commands related to picking. It then instructs the high-level controller to perform pick-place operations by calculating the inverse kinematics for each joint. The end effector in this study is a pneumatic actuator for the case that does not utilize pose estimation and a mechanical gripper for the case that introduces pose estimation, respectively.

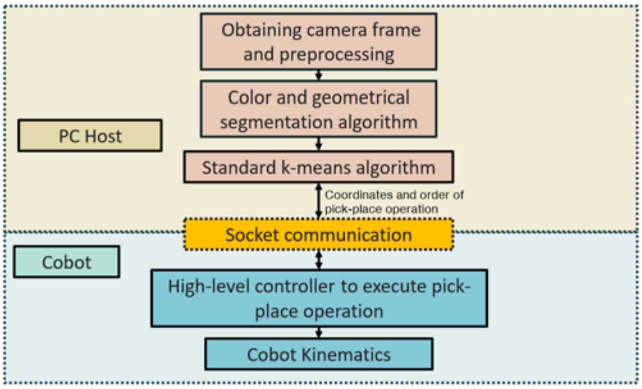



**Figure 2** The proposed methodology and its associated diagram.

a. Computer vision: Preprocessing and segmentation algorithm

Before segmentation, denoising techniques aim to eliminate unwanted noise from images, enhancing the data quality and improving the accuracy of subsequent steps. In this study, noise reduction is performed to eliminate various types of noise, mainly Gaussian noise, ensuring that the input data is more reliable, leading to more precise and consistent segmentation results. Next, the image segmentation algorithm divides the input image into meaningful regions or segments, thereby identifying distinct objects or regions of interest. This initial segmentation step helps distinguish objects from the background and separate them for further analysis.



DOI: 10.60101/jarst.2023.254153

Here, the algorithm of the host computer primarily relies on OpenCV, a widely utilized computer vision library, for color segmentation to locate a specific-colored object captured by a camera. The filtered image is converted to the HSV color space, enabling the separation of color information from brightness. A mask is then created by setting lower and upper HSV thresholds, effectively isolating pixels with color of interest. The findContours function detects the contours of these specific pixels, focusing solely on the outer contours. The calculated value is compared to a predetermined threshold to assess if the object closely matches the desired shape to ensure further shape validation. For instance, one of the shape parameters is calculated using circularity, $C = 4\pi A/P^2$, where $A$ is the contour area, and $P$ is the contour perimeter, respectively. For the case of a spherical object, $C$ is close to 1.

In addition, to detect the location and estimate the pose of non-circular, for instance, rectangular objects, the minAreaRect function in OpenCV is utilized. The minAreaRect function in OpenCV is used to find the minimum area rectangle that encloses a set of points. This methodological approach involves tracing a continuous and unbroken curve that precisely outlines the spatial boundaries of the square under examination. It can detect and localize non-circular objects, such as rectangles or bounding boxes of the objects in an image. The center, width, and height of the rectangular object can be identified, as well as the decision of the picking location of the objects. In terms of orientation, the object's rotation angle is also utilized. This is a particular case so that the $6^{th}$ joint of the cobot can be adequately rotated prior to picking the objects according to the z-axis rotation (Rz). The Rz transformation benefits the robot arm that uses a gripper to pick up objects. Further, this can also perform pose estimation, determining the position and orientation of identified objects.

These two segmentation algorithms are used to identify the objects' types, shapes, locations, orientations, and poses. With slight tuning of parameters, the algorithm can identify the objects' properties. For instance, contours with circularity above the threshold can differentiate one object from another or the rectangular object with its associated pose. Those segmented images will then be used to split the data into small clusters of images. Subsequently, the objects are grouped into data clusters that will be sent as input to the k-means clustering algorithm.

b. K-means algorithm

The k-means algorithm is employed to perform clustering of the objects, deciding the order of operation into subtasks using the L-2 norm. Following the segmentation of the image, the k-means algorithm is subsequently employed. K-means is an unsupervised machine-learning algorithm commonly used for clustering tasks. The k-means clustering algorithm is notable for its simplicity, accessibility to implementation, and efficiency, as it can process large datasets that can also be scaled. In addition, it is incredibly versatile and can be used with varying scales and dimensions. However, k-means clustering has drawbacks as the algorithm is highly sensitive to the predefined k values, which is challenging to determine. Outliers also impact the results, which necessitate pre-processing before applying k-means clustering to maintain accuracy. One of the main disadvantages of the algorithm is that it is highly susceptible to converging to local minima, posing a challenge in cases where its data, especially for each complex cluster, is not well-separated. In object localization, the k-means algorithm helps in grouping similar pixels. By iteratively assigning pixels to different clusters and optimizing their centroids, the k-means algorithm is also applied to the data to find the characteristics of each object and perform clustering. The k-means algorithm is an algorithm that groups the given data into $k$ clusters. When this algorithm receives $n$ objects, it is divided into $k$ groups that are less than or equal to $n$, and each group forms a cluster. Here, $k$ is determined from image segmentation acquired from *Section a*.

Each cluster represents a centroid of potential pick-and-place location. The algorithm iteratively changes the location of the centroid, and it runs until the centroids converge. In other words, to achieve optimal clusters, the variance in the distance difference between each cluster and the data within that cluster must be minimized [24, 25]. The objective function of k-means is written as:

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left|x_i - \mu_j\right|^2 \left\|x_i - \mu_j\right\|^2 \tag{1}$$

where $i$ iterates over the data points, j iterates over the clusters, $x_i$ is a data point, and $\mu_j$ is the centroid of the $j$-th cluster. The $|x_i - \mu_j|$ represents the Euclidean distance between a data point and the centroid. $k$ is number of clusters.

c. Combined algorithm to obtain order of operation

The combined algorithm is utilized to send the commands the cobot to decide the order of operation for picking and placing objects, streamlining its manipulation and placement tasks. Combining image segmentation and the k-means algorithm is essential to indicate precise object localization. First, the image segmentation algorithm identifies potential objects



DOI: 10.60101/jarst.2023.254153

or regions of interest in the scene. Then, the k-means algorithm refines this localization by grouping together pixels with similar characteristics and indicating the objects' location. Snippets of pictures are used in the k-means algorithm for segmentation to better highlight the features of the object by separating distinct clusters representing different features within the image frame. The proposed algorithm is designed to impose fewer restrictions on the color segmentation threshold. K-means clustering is less affected by specific color values, as it focuses on overall color distribution. Combining these two algorithms improves object segmentation accuracy, mainly when color segmentation falls short.

Finally, the $L^2$ or Euclidean norm is employed to measure the distance between the center of the robot gripper and the centroid of each cluster. The process determines the object that has the minimum norm based on the picking location and the gripper location. This prioritization of sequence allows the cobot to minimize travel time by optimizing the path to be the shortest distance of operation. This approach aims to improve the accuracy and reliability of object localization significantly, making subsequent tasks like object recognition, tracking, and pick-and-place operations much more efficient. The order of pick-and-place operations and coordinate and pose estimations are then transmitted to the cobot software using socket communication.

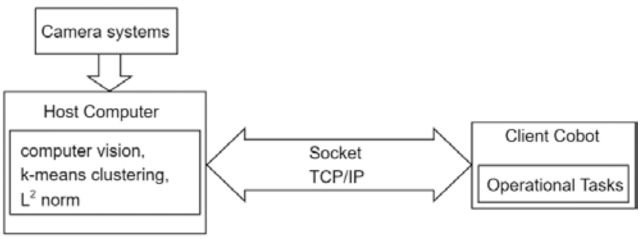



**Figure 3** TCP/IP socket communication between host computer and client cobot (adapted from [26, 27]).

d. Socket communication to cobot

The communication process starts with establishing TCP/IP as a communication channel to transmit data between the host computer and the cobot, as illustrated in Figure 3, through a Local Area Network (LAN) to ensure the integrity and security of the data transmission. Specifically, TCP/IP socket communication is chosen due to its established reliability and broad compatibility across diverse devices [15, 26, 27]. The host is responsible for creating an endpoint that clients can connect to while the client component initiates the connection to the server. The TCP/IP socket connection facilitates the data exchange and instructions between the server computer and the client cobot. The server processes visual information, generates commands, and transmits them to the client cobot, RobotStudio, enabling the cobot to execute tasks based on the computer vision analysis results obtained from the server.

The main objective of this communication setup is to enable the transfer of command information from the server computer, where computer vision and pick-place operation tasks are executed, to the cobot client for subsequent execution. Socket communication was selected as the communication mechanism in this study to facilitate information exchange between the host computer and the client IP ports. The client-server setup distinguishes a clear division of responsibilities: the client, or host computer in this case, initiates communication tasks and sends data, while the server, represented by the cobot, awaits incoming connections and processes the requests it receives.

The host computer initializes a socket communication to transmit the data in a binary socket package, including the location coordinates to perform pick place, pose, and object orientation to the Robot Studio. The send() and recv() functions transmit and receive data between devices. The Robot Studio, utilizing the RAPID programming language designed for ABB industrial robots and cobots, receives and interprets the control data, facilitating the command and regulation of cobots' positional adjustments and operation. The cobot then sends the signal back once the data is interpreted and performed as the data is sent accordingly. This coordinated control framework facilitates the cobot to navigate precisely to the specified location and execute the requisite actions to grasp the targeted object successfully.

e. High-level controller to execute pick-place operations and cobot kinematics.

To perform precise and accurate manipulation of the designated object, the high-level controller within a cobot framework functions as the processing task. It receives and interprets commands from the host via TCP/IP. These commands serve as precise target coordinates within the operational workspace of the cobot. The high-level controller's primary responsibility entails the formulation and execution of a trajectory plan constrained by kinematic limitations and spatial considerations. The trajectory planning focuses on devising efficient and obstacle-avoidant trajectories for the end-effectors of the cobots, enabling it to reach specified target positions within its workspace. In low-level operation, inverse kinematics entails the mathematical computation of joint angles required to position the end-effector of the cobot precisely at a designated location and orientation. It serves as the



DOI: 10.60101/jarst.2023.254153

connection between high-level task objectives and the joint space of the cobot, facilitating precise control of its movements. Subsequently, the controller activates the end-effector system to complete the pick-place operations.

During this stage, the controller monitors sensory feedback, particularly force-related data, to ensure objects' secure and accurate manipulation. In addition, collision detection is pivotal in ensuring safety by continuously monitoring the environment, identifying potential collisions with obstacles or other objects, and adjusting the trajectory or stopping its motion to prevent accidents and equipment damage. After a sequence of pick-place operations is completed, the information is sent back to the host and waits for the subsequent tasks.

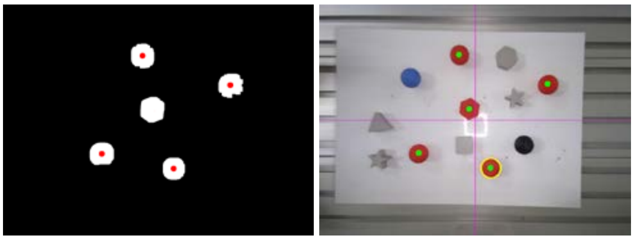
The proposed methodology of this study is summarized as follows:

1. Obtain camera frame and perform preprocessing.
2. Perform image segmentation, which includes:
   a. Color thresholding, clustering of contours using a geometrical approach.
   b. K-means clustering based on a.
   c. combining a and b to identify each cluster.
3. Determine the order of pick-place operations of the cluster based on $L^2$ norm.
4. Transmit the localized data to the cobot to perform pick-place operations.
5. Execute pick-place operations and wait for confirmation of the operation from the cobot before repeating the process from step 1.

## RESULTS AND DISCUSSIONS

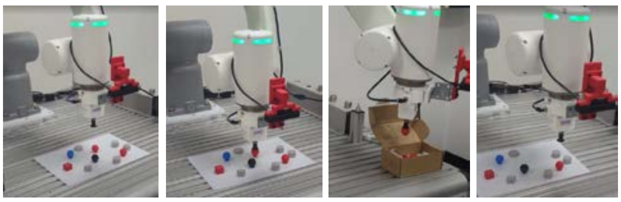
This study randomly positions various objects with different colors (blue, grey, red, black) and shapes (stars, rectangles, cylinders, spheres, hexagons) within the workspace. This study specifically focuses on demonstrating the accuracy of identifying red spherical and rectangle objects to assess the proposed algorithm. The host computer, equipped with a proposed algorithm, successfully determines the correct positions of the indicated objects and can locate the closest one concerning the location of the pneumatic suction. A validation process is conducted to evaluate the precision of the pick-and-place operations by assessing the performance of picking up the designated red spheres. A total of 500 pick-and-place operations with a speed of 100 mm/s are performed, and each operation is categorized as either a correct or incorrect pick-up. The computer vision algorithm is initially applied without k-means clustering, resulting in an observed accuracy of 87.4%.

After that, the proposed algorithm is implemented to detect the positions of the targeted red spheres during the validation procedure. Changes have significantly influenced the system's accuracy in lighting conditions and pick-place operating speed. Additionally, through further analysis, the sensitivity of the proposed algorithm can be fine-tuned by adjusting the threshold of location information within each frame window of capture. In this study, the algorithm was applied to every 15 consecutive frames from the camera to identify the optimal location of pick-and-place operations, resulting in an average response time of 0.52 ± 0.1 seconds for each pick-place operation. The cobot successfully executes the pick-and-place tasks with high accuracy, picking up the red spheres based on the identified coordinates and achieving an accuracy rate of 98.2% (See Figure 4). Image segmentation and clustering can accurately identify the number and location of clusters. The proposed algorithm performs better than the one without incorporating k-means clustering. The pick-place operation is illustrated in Figure 5. All the errors were associated with the proposed localized algorithm, which sometimes could not identify the objects from the captured images.



(a) (b)

**Figure 4** A snippet frame from the algorithm (a) after image segmentation and clustering of k-means algorithm and (b) resulting location of the picking operation.



(a) (b) (c) (d)

**Figure 5** A pick-place operation process (a,b,c,d) with pneumatic suction on specified objects of interest.

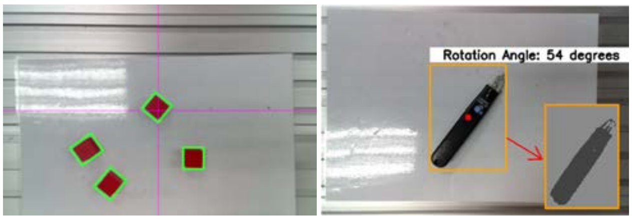
Variations in lighting conditions, occlusions, or challenging shapes may also have contributed to this source of error. In conditions with reduced



DOI: 10.60101/jarst.2023.254153

lighting, specifically at 1/3 of the standard lighting power, the accuracy of the pick-place operation is reduced to 67.1%. The primary factor contributing to the reduced accuracy is the inaccuracy of the color segmentation, accounting for 35.2% of the errors. Under diminished lighting conditions, the pick-and-place operation struggled to accurately identify the correct color range, leading to a decline in overall performance. Despite this limitation, the k-means clustering portion of the algorithm proved to be extremely helpful in object localization, enabling the cobot to successfully grasp objects even though they may possess the incorrect color or shape from error in segmentation. Another error occurs when the operating speed of the cobot is increased. The average response time in this study can be reduced through further optimization of the framework, thereby mitigating errors at higher jogging speeds of the cobot arms.

Another study involves object detection and pose estimation used along with machine vision to extract relevant information, such as key points for the end effector to grasp the specific part of the object precisely. Here, the pick-place operation of the rectangular object with a mechanical gripper is demonstrated. Once the cobot receives the coordinate data and performs inverse kinematics of the joint angles, the resulting angle of the object pose is sent to the cobot. The rotation of the mechanical gripper is performed to precisely place the cobot on the specific location and orientations concerning the workpiece. This demonstration is performed with the pick-place operations of the rectangular object, as seen in Figure 6 (a). The accuracy of this study is found to be 94% from a total of 200 operations at a speed of 100 mm/s, slightly lower than the study without pose estimation. This application creates a framework for precise object detection, which is crucial in applications like manufacturing and automation. Further applications of this study can be applied to critical points and image localization across a broad spectrum of industries. For instance, object detection instead of image segmentation can be computationally intensive to obtain the preliminary position of the objects of interest. K-means clustering is then applied to the image frame to obtain a more refined feature and segment the object boundaries for the pick-place operations. In this preliminary study, it is found that using You Only Look Once (YOLO) v8 [28], which is a well-known efficient convolutional neural net object detection along with k-means clustering of the image frame, could be used to obtain the exact position and pose of the object of interest for grasping accordingly. Figure 6 (b) shows that the proposed implementation starts with object detection to identify the local coordinates of the object of interest. The algorithm is then passed on to the k-means clustering, where the segmentation occurs, resulting in feature extraction and pose estimation for pick-place operations.




(a) (b)

**Figure 6** A snippet frame from the algorithm with pose estimation for the mechanical gripper.

## CONCLUSIONS

This study proposes an approach to pick-and-place operations using computer vision. Integrating the computer vision algorithm allows the cobot to perceive and understand its surroundings by extracting relevant visual information. Utilizing the k-means clustering algorithm complements the computer vision component by offering a structured depiction of the localized objects, which are fundamental for accurate pick-and-place operations. The host computer then establishes a TCP/IP communication channel and transmits data to a cobot through a socket connection, enabling real-time command data exchange. The experimental results demonstrate the effectiveness of this integration, showcasing the enhanced adaptability and efficiency of pick-and-place operations in cobots. This research performs pick-and-place operations with 98.2% accuracy, achieving an average response time of $0.52 \pm 0.1$ seconds, compared to the original algorithm of 87.4% accuracy. Further study that includes pose estimation in pick-and-place operations demonstrates 94% accuracy. The study emphasizes the viability of employing a k-means algorithm in addition to image segmentation to execute pick-and-place operations by utilizing socket communication, enabling seamless data transfer between the cobot and the software, thereby enhancing overall efficiency.

## ACKNOWLEDGEMENTS

This research is financially supported by Rajamangala University of Technology (grant number D04/2566).

## REFERENCES

1. Preuveneers D, Ilie-Zudor E. The intelligent industry of the future: A survey on emerging trends,





DOI: 10.60101/jarst.2023.254153

research challenges and opportunities in Industry 4.0. J Amb Intel Smart En. 2017;9(3):287-98.

2. Kadir BA, Broberg O, Souza da Conceição C. Designing human-robot collaborations in industry 4.0: explorative case studies. In: *DS* 92: Proceedings of the DESIGN 2018 15th International Design Conference. 2018 May 21-24; Dubrovnik, Croatia. Dubrovnik: Design Society; 2018. p. 601-10.

3. Krüger M, Wiebel CB, Wersing H. From Tools Towards Cooperative Assistants. In: Proceedings of the 5th International Conference on Human Agent Interaction; 2017 Oct 17-20; Bielefeld, Germany. ACM; 2017. p. 287-94.

4. Li A, Gurocak H. Modified Bug Algorithm with Proximity Sensors to Reduce Human-Cobot Collisions. In: 2023 9th International Conference on Automation, Robotics and Applications (ICARA). 2023 Feb 10-12; Abu Dhabi, United Arab Emirates. Abu Dhabi: NYU Abu Dhabi; 2023. p. 80-5.

5. Syyrilä T. Development of a Collaborative Robot Manufacturing Workstation. [Master's Thesis]. Mechatronic System Design: Lahti University of Technology LUT; 2022.

6. Zhu Z, Hu H. Robot learning from demonstration in robotic assembly: A survey. Robotics. 2018; 7(2):17.

7. Peternel L, Petrič T, Babič J. Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. Autonomous Robots. 2018;42:1-17.

8. Borboni A, Reddy KVV, Elamvazuthi I, AL-Quraishi MS, Natarajan E, Azhar Ali SS. The expanding role of artificial intelligence in collaborative robots for industrial applications: A systematic review of recent works. Machines. 2023;11:111.

9. Biggs G, MacDonald B. A survey of robot programming systems. In: Proceedings of the Australasian conference on robotics and automation. 2003. p. 1-3.

10. Hughes C, Hughes T. Robot programming: a guide to controlling autonomous robots. Vol. 1. Indianapolis, Indiana:Que; 2016.

11. Rossano GF, Martínez C, Hedelind M, Murphy S, Fuhlbrigge T. Easy robot programming concepts: An industrial perspective. In: 2013 IEEE International Conference on Automation Science and Engineering (CASE). 2013 Aug 17-20; Wisconsin, USA. IEEE; 2013. p. 1119-26.

12. Pedersen MR, Nalpantidis L, Andersen RS, Schou C, Bøgh S, Krüger V, et al. Robot skills for manufacturing: From concept to industrial deployment. Robotics and Computer-Integrated Manufacturing. 2016;37:282-91.

13. Manoharan M, Thilak KSS, Kumaraguru S. Collaborative motion synchronization for affordable open-source cobots. In: 2022 IEEE 6th Conference on Information and Communication Technology (CICT); 2022 Nov 18-20; Gwalior, India. IEEE; 2022. p.1-5.

14. Iriondo A, Lazkano E, Susperregi L, Urain J, Fernandez A, Molina J. Pick and place operations in logistics using a mobile manipulator controlled with deep reinforcement learning. Appl Sci. 2019; 9(2):348.

15. Andhare P, Rawat S. Pick and place industrial robot controller with computer vision. In: 2016 International Conference on Computing Communication Control and automation (ICCUBEA). 2016 Aug 12-13; Pune, India. IEEE; 2016. p. 1-4.

16. Migimatsu T, Bohg J. Object-centric task and motion planning in dynamic environments. IEEE Robotics and Automation Letters. 2020;5:844-51.

17. Stulp F, Fedrizzi A, Beetz M. Learning and performing place-based mobile manipulation. In: 2009 IEEE 8th International Conference on Development and Learning; 2009 Jun 5-7; Shanghai, China. IEEE; 2009. p. 1-7.

18. Soltan S, Oleinikov A, Demirci MF, Shintemirov A. Deep learning-based object classification and position estimation pipeline for potential use in robotized pick-and-place operations. Robotics. 2020;9(3):63.

19. Kumar R, Kumar S, Lal S, Chand P. Object Detection and Recognition for a Pick and Place Robot. In: Asia-Pacific World Congress on Computer Science and Engineering; 2014 Nov 4-5; Nadi, Fiji. IEEE; 2015.

20. Lobbezoo A, Qian Y, Kwon HJ. Reinforcement learning for pick and place operations in robotics: A survey. Robotics. 2021;10(3):105.

21. Zhou W, Meng F, Li K. A cherry tomato classification-picking robot based on the K-means algorithm. J Phys: Conf Ser. 2020;1651:012126.

22. Anggi Maharani D, Fakhrurroja H, Riyanto R, Machbub C. Hand gesture recognition using K-means clustering and Support Vector Machine. In: 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE); 2018 Apr 28-29; Penang, Malaysia. IEEE; 2018. p. 1-6.

23. Wakayama T, Garcia Ricardez GA, Hafi LE, Takamatsu J. 6D-Pose Estimation for Manipulation in Retail Robotics using the Inference-embedded



DOI: 10.60101/jarst.2023.254153

OAK-D Camera. In: 2022 IEEE/SICE International Symposium on System Integration (SII). 2022 Jan 9-12; Narvik, Norway. IEEE; 2022. p. 1046-51.

24. Abernathy A, Celebi ME. The incremental online k-means clustering algorithm and its application to color quantization. Expert Systems with Applications. 2022;207:117927.
25. Ezugwu AE, Ikotun AM, Oyelade OO, Abualigah L, Agushaka JO, Eke CI, et al. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence. 2022;110: 104743.
26. Hui-ping L, Dai-min C, Miao Y. Communication of multi-robot system on the TCP/IP. In: 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC). 2011 Aug 19-22; Jilin, China. IEEE; 2011. p. 1432-35.
27. Shin S, Yoon D, Song H, Kim B, Han J. Communication system of a segmented rescue robot utilizing socket programming and ROS. In: 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). 2017 Jun 28-Jul 01; Jeju, Korea (South). IEEE; 2017. p. 565-9.
28. Jocher G, Chaurasia A, Qiu J. YOLO by Ultralytics (Version 8.0.0) [Computer software] [Internet]. GitHub, Inc, 2023. [cited 2023 Sep 10]. Available From: https://github.com/ultralytics/ultralytics/blob/main/CITATION.cff



DOI: 10.60101/jarst.2023.254153