IJRCS

International Journal of Robotics and Control Systems
Vol. 5, No. 1, 2025, pp. 440-458
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Efficient Vision-Guided Robotic System for Fastening Assembly Using YOLOv8 and Ellipse Detection in Industrial Settings

Yan Kai Tan [a,1], Kar Mun Chin [a,2], Yeh Huann Goh [a,3,*], Tsung Heng Chiew [a,4],
Terence Sy Horng Ting [a,5], Ge MA [b,6], Chong Keat How [a,7]

[a] Department of Mechanical Engineering, Faculty of Engineering And Technology, Tunku Abdul Rahman University of Management and Technology Kuala Lumpur, Malaysia
[b] Guangzhou University, Guangzhou, 510006, People's Republic of China
[1] tanyk-wg17@student.tarc.edu.my; [2] chinkm@tarc.edu.my; [3] gohyh@tarc.edu.my; [4] chiewth@tarc.edu.my;
[5] terencetsh-wp18@student.tarc.edu.my; [6] m_ge@gzhu.edu.cn; [7] vichck2020@gmail.com
* Corresponding Author

ARTICLE INFO

Article history
Received November 11, 2024
Revised December 26, 2024
Accepted January 13, 2025

Keywords
Robotic Vision-Based Assembly;
YOLOv8 Object Detection;
Ellipse Detection;
3D Positioning;
Depth Camera

ABSTRACT

The assembly of fastening components traditionally relies on labour-intensive human-machine collaboration, which incurs high costs. Existing methods often assume fixed positions or use markers for guidance, requiring extra effort to place and maintain them. This study aims to develop an intelligent control system for a vision-equipped robotic arm to autonomously assemble fastening components in industrial settings, enhancing flexibility and reducing labour costs. The system integrates object detection with edge and ellipse detection, alongside filtering techniques, to accurately locate the centres of the fastening components. The key contribution is the system's ability to perform autonomous assembly without predefined positions, enhancing flexibility in varied environments. YOLOv8 is employed to detect the bolt and nut, followed by edge and ellipse detection to pinpoint centre coordinates. A depth camera and kinematic calculations enable accurate 3D positioning for pick-and-place tasks. Experimental results demonstrate the system's high effectiveness, with less than 1% of targets undetected. Based on experiments conducted in randomly arranged conditions, the system demonstrated high effectiveness, achieving over 99% detection accuracy. It achieved an 87% average success rate for picking fastening components ranging from sizes M6 to M18, and a 90% success rate for precise placement. Additionally, the system demonstrated robustness across various component sizes, with a minor increase in orientation errors for smaller components, attributed to depth estimation challenges. Future work could explore alternative depth data collection methods to improve accuracy. These results confirm the reliability of the system in flexible assembly tasks, demonstrating its potential to reduce costs by minimising manual involvement in industrial settings.

http://dx.doi.org/10.31763/ijrcs.v5i1.1705

1. Introduction

Bolts and nuts are common fasteners widely used across the engineering field. In most industries, bolt and nut assembly is typically carried out by humans using tools such as wrenches or tightening machines [1]. When both humans and machines are involved, this process is referred to as human-machine collaboration (HMC). HMC enhances the quality and efficiency of the assembly. However, this approach depends heavily on skilled operators, presenting a significant challenge. Consequently, there is a growing need to reduce the reliance on human operators in bolt and nut assembly. With the gradual advancements in robotic arm technology, this technology offers a promising solution to this challenge.

Robotic arms are commonly used for repetitive tasks. These tasks include picking and placing objects in fixed positions. They are valued for their precision, consistency, and ability to operate continuously without fatigue. However, there is increasing interest in developing compliant robotic arms that can adapt to external forces or changes in their environment [2], [3]. Compliant robotic arms offer advantages such as adaptability to unstructured environments and the ability to handle delicate tasks with precision. For instance, one study proposed the use of two industrial robots to perform the bolt and nut assembly process through force feedback [2]. Another approach relies on the kinematic model of a dual-arm robotic system to handle bolt and nut assembly [4]. This method eliminates the need for a force sensor. However, this method requires precise control over the positions of both the gripped bolt and nut. A further study introduced human-robot collaboration to accomplish the assembly, replacing one of the two human operators with a robot [5]. Additionally, an automated process for assembling Hi-Lite bolts and collars using a robotic arm has been proposed [6].

As discussed above, most research and proposed methods for bolt and nut assembly focus on the connection process. These methods typically assume that the bolt and nut are in fixed, predefined positions to ensure accuracy and efficiency. However, fixed positions are often impractical in real-world scenarios where placement errors may cause variations in the locations of these components. Therefore, instead of relying on predefined positions, alternative methods are required to recognise and locate the bolts and nuts. One such method involves applying markers to the target object for recognition. For example, humanoid robots are used for bolt and nut assembly in aircraft production environments [7]. A fiducial marker system has been employed to enable the vision system to detect the target's position accurately [8]. This ensures precise grasping and manipulation tasks in bolt and nut assembly. Predefined positions and markers have led to inefficiencies and increased costs. Extra work is required to place the marker precisely on the bolt and nut and position them accurately.

According to research by Xu et al. [1], three main technologies are essential when determining the 3D position of bolts and nuts: image segmentation, target recognition, and 3D position calculation. Image segmentation involves methods such as thresholding [9]–[11]. In thresholding, the target is separated based on a set threshold, using objective functions like interclass variance [12] and interclass variance with correlation [13]. In multi-threshold segmentation, techniques such as the improved firefly algorithm [14], whale optimisation algorithm, and moth-flame optimisation [15] have been proposed to identify appropriate thresholds. Another approach, edge segmentation, extracts edge data from images [16]–[19]. Region-based methods like Simple Linear Iterative Clustering (SLIC) [20]–[24] are also used for segmentation. SLIC is a popular image segmentation algorithm that clusters pixels into superpixels, reducing the complexity of subsequent image processing tasks. Clustering analysis methods, including the K-means algorithm, fuzzy theory, and Convolutional Neural Networks (CNNs), can further enhance image segmentation [25]–[29].

Target recognition relies on features such as colour, texture, and shape [30]–[32]. Local feature recognition methods include Scale-Invariant Feature Transform (SIFT) [33], Speeded-Up Robust Features (SURF) [34], and Oriented FAST and Rotated BRIEF (ORB) [35]. Statistical methods such as Support Vector Machine (SVM) [36], Bayesian approaches [37], decision-making algorithms, and neural networks [38], [39] are also employed. Additionally, semantic segmentation methods have been introduced to improve the accuracy of recognition systems [40]–[42]. 3D position calculation typically depends on depth cameras. In this paper, a depth camera with binocular vision is employed. It uses two cameras positioned at different locations to estimate depth data [43]–[45], a technique commonly known as stereoscopic technology. Building on the depth data provided by the stereoscopic camera, kinematic calculations play a critical role in ensuring the precise movement of the robotic arm. Forward kinematics is used to determine the position and orientation of the arm's end effector based on joint angles. Inverse kinematics calculates the required joint angles to achieve a desired position and orientation. Together, these calculations enable the system to accurately perform pick-and-place tasks in dynamic and unstructured environments.

Several studies similar to this research are introduced here. First, a robotic system was developed to perform autonomous bolt and nut assembly tasks during hot-line maintenance [46]. Additionally, a collaborative robot for wheel hub bolt and nut assembly was proposed. It aimed to improve the quality of the assembly process, reduce cycle times, and address ergonomic issues associated with manual nut assembly [47]. Another study presented a method for object recognition to assist a robotic system. The system autonomously performed maintenance tasks on power distribution lines, including screwing a nut onto a bolt [48]. In these projects, the workpieces, such as bolts or nuts, had to be fixed in specific positions to facilitate the assembly process. Furthermore, projects that required human guidance were not fully automated. As a result, they did not significantly reduce the need for manual labour. By contrast, this study proposes a vision-guided control system that eliminates the need for predefined positions or markers. This enables full automation and adaptability in dynamic industrial settings.

A separate method was proposed to reduce positioning errors in high-precision bolt and nut assembly by combining visual and force feedback [49]. In this method, the bolt and nut positions did not need to be predefined. No human guidance was required, making it similar to the method proposed in this paper. However, the key difference lies in the 3D position calculation of the bolt and nut. In the method presented [49], the camera was moved to four positions above the bolt and nut. The depth data collected from these positions was then averaged to determine the distance between the depth camera and the components. In contrast, the proposed method in this paper requires the depth camera to be positioned once. It is placed directly above the bolt and nut in close view to calculate the distance. When the proposed method places the nut close to the bolt, peg-in-hole mating technology can be applied to ensure proper mating during the assembly process [1]. Following this, a tightening tool can be implemented alongside the proposed method. A tightening control system can also be added to ensure the quality of the tightening process [3], [50]–[52].

From the reviewed literature, it is evident that existing methods for bolt and nut assembly are effective under controlled conditions. These conditions rely on predefined positions or markers. These methods often rely on fixed positions or human guidance, limiting their applicability in dynamic and unstructured environments. Furthermore, some studies employ advanced techniques like semantic segmentation and object recognition. However, they lack precision in determining the exact centre of the fasteners, which is critical for autonomous assembly tasks [53]–[55]. These challenges highlight the need for a system that can adapt to varied conditions, eliminate reliance on predefined positions, and ensure precise centre detection. This study aims to address these gaps by proposing a vision-guided control system integrating YOLOv8 and ellipse detection.

This study aims to develop an intelligent control system for a vision-equipped robotic arm. The system autonomously assembles bolts and nuts in industrial settings, enhancing flexibility and reducing labour costs. Unlike previous approaches, the developed system does not rely on predefined positions or markers. The proposed method was validated through experiments conducted in dynamic and randomly arranged conditions. It achieved over 99% detection accuracy and high success rates in picking and placing nuts onto bolts. This approach is particularly suited for applications requiring high precision in detecting and positioning bolts and nuts. It is especially useful in dynamic environments where predefined positions and markers are impractical. However, the system's computational complexity and reliance on advanced hardware may present challenges in resource-constrained settings. This highlights a trade-off between accuracy and implementation cost. In this study, YOLOv8 and ellipse detection were combined to detect the centre of bolts and nuts. YOLOv8 excels at detecting the presence of bolts and nuts due to its fast detection speed, making it suitable for real-world applications [56]–[58]. However, the YOLOv8 model has a limitation in accurately detecting the centre of bolts and nuts. The centre of its detected boundary box does not always align with their actual centre. Both bolts and nuts share a circular shape feature, with the centre of the circle representing their actual centre. To address this limitation, ellipse detection was employed for precise centre detection [59]. While ellipse detection is highly accurate in determining the centre of bolts and nuts, it cannot independently detect their presence. It may also incorrectly identify all possible ellipses as bolts or nuts.

The research contribution is the development of a vision-guided system that eliminates the need for predefined positions or markers, enabling assembly in dynamic environments. It also achieves high detection accuracy and precise centre localisation of bolts and nuts, addressing the limitations of existing methods.

2. Methodology

2.1. Overall Method

The overall process of the proposed method is depicted in Fig. 1. The proposed system integrates a depth camera, a robotic arm, and a control system to achieve autonomous bolt and nut assembly. The depth camera is capable of capturing real-time 3D spatial data to localise bolts and nuts in dynamic environments. This data is fed into the control system, which uses YOLOv8 for initial object detection and ellipse detection to accurately locate the centre of each bolt and nut. The control system then calculates their 3D positions and uses inverse kinematics to generate precise movement commands for the robotic arm. The robotic arm, guided by these commands, performs pick-and-place operations with high accuracy using KUKA Ethernet KRL for computer-robot communication.

The process begins with the depth camera capturing an image, producing two outputs: a colour image and depth data. The colour image is first subjected to object detection to identify the component and its bounding box. The image is then cropped based on the bounding box to reduce computational load and minimise the chances of false centre estimations. If no bolt or nut is detected, the system will wait until a bolt or nut is detected.

Next, the cropped image undergoes Canny edge detection, an algorithm that detects edges based on specified lower and upper thresholds applied to a greyscale image. Canny edge detection is chosen for its superior accuracy compared to other edge detectors, which may fail to identify all significant edges or struggle with low-resolution images. Its robustness to noise and ability to localise edges precisely make it particularly well-suited for detecting the circular features of bolts and nuts in dynamic industrial environments. In this application, the key edges are the circular edge of the bolt and the inner circular shape of the nut.

After edge detection, contour detection is performed on the edge image using the findContours function from OpenCV, which identifies all connected edges as contours. The next step involves fitting each detected contour into an ellipse using OpenCV's fitEllipse function, which employs the least squares method. Ellipses are preferred over circles, as circles can become distorted depending on their position in the image. Although this distortion is typically minimal, using ellipses ensures that all significant contours are accurately detected. Both bolts and nuts have circular features, and these can be used to locate their centre points.

Since multiple ellipses may be detected, an ellipse filter is applied to identify the most suitable ellipse representing the bolt or nut. Once the ellipse corresponding to the target object is identified, the image coordinates of its centre point are recorded and used to extract depth data from the depth image. Using the depth data of the centre point, the transformation from the camera to the target object is calculated. Finally, with the known rotation angles of all the robot joints, the position of the target object relative to the robot base is determined using forward kinematics.

Overall, the system relies on advanced hardware, including a depth camera, a high-performance computer, and a robotic arm with precise control capabilities, which may increase the initial setup cost. However, these costs are justifiable in industries that prioritise high precision and automation to reduce labour expenses and enhance productivity. For instance, the proposed method can be applied to automotive assembly lines, where bolts and nuts must be installed on randomly positioned components, enabling efficient and accurate operations in dynamic environments.

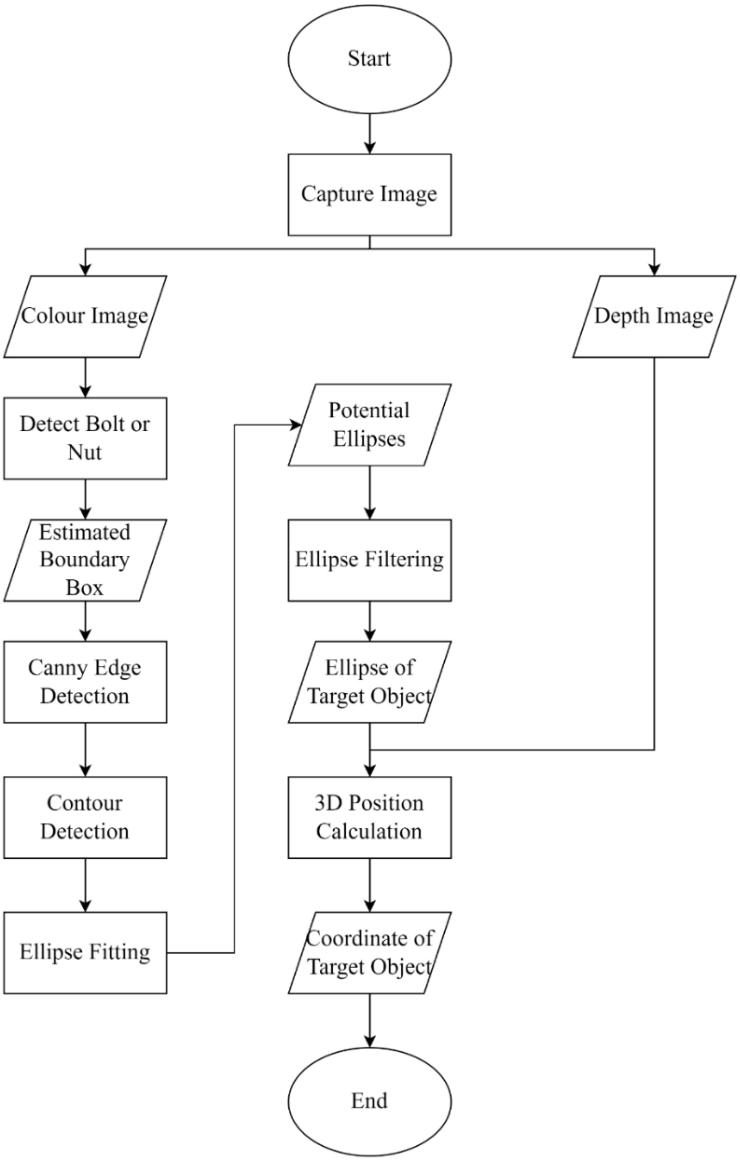


Fig. 1. Flowchart illustrating the overall methodology of the proposed system for bolt and nut assembly. The process includes data capture with a depth camera, object detection using YOLOv8, edge detection, ellipse fitting for centre localisation, and 3D spatial coordinate calculation to guide the robotic arm in performing precise pick-and-place operations

2.1.1. Bolt and Nut Centre Point Detection

To localise the bolt and nut centre point in the workspace, relying solely on image segmentation and target recognition techniques was insufficient, particularly in noisy environments. Therefore, an object detection algorithm was applied to accurately localise the position of the target object. In this paper, the YOLOv8 object detector was selected for detecting the bolt and nut in the workspace due to its fast detection speed, high accuracy and availability on the Ultralytics platform [60]. YOLOv8 is classified as a one-stage object detector, primarily designed for fast detection. In contrast, two-stage object detectors are typically focused on achieving higher accuracy in object prediction. However, YOLOv8 stands out among one-stage detectors by delivering fast detection speeds without compromising detection accuracy. According to Ultralytics, YOLOv8 outperforms YOLOv7 and earlier versions on the COCO dataset. Among the different YOLOv8 architectures: YOLOv8x, YOLOv8l, YOLOv8m, YOLOv8s, and YOLOv8n, the larger models achieve higher mean average precision on the COCO dataset. However, research shows that all YOLOv8 models perform similarly when detecting small bolts within a range of 20 cm to 60 cm [61]. Due to its fast inference and training speed, the YOLOv8n model was selected for training the bolt and nut detector in this paper.

The training and validation datasets were prepared with various backgrounds to ensure that the bolt and nut characteristics could be learned in different environments. Backgrounds included a white surface, metal texture, and a chessboard pattern with black and white squares. A total of 224 images with a resolution of 1920 × 1080 pixels were captured to create the dataset. Images were also captured at two different distances between the table and the camera, approximately 150 mm and 250 mm, to ensure the detector could learn and localise the bolt and nut at varying distances between the camera and workspace. No data augmentation techniques were applied to the images to maintain their original characteristics. This was essential as the object detector needed to detect the bolt and nut from different distances for the picking and placing tasks.

In this paper, the bolt and nut were always placed with their circular surfaces parallel to the workspace and camera frame. This assumption was made because the circular surface is typically parallel to the connecting components in industrial assembly processes, simplifying detection and localisation. The centre of the bolt and nut corresponded to the centre of these circular shapes. Therefore, the annotation of the bolt focused on the circular shape of the bolt point during object detector training. However, the nut annotation included the entire nut, as the threaded part of the bolt could introduce noise by appearing as a circular arc from certain angles. Detecting the bolt point proved challenging because the object detector struggled to identify the circular edge precisely. As a result, the centre of the detected boundary box did not always align with the bolt point's centre.

To address this issue, an image processing method is proposed to detect the precise centre of the bolt or nut. First, Canny edge detection was applied to detect edges. One challenge with Canny edge detection is setting the threshold values. In this method, the median value of the greyscale image, cropped based on the boundary box, was calculated. The lower threshold was set at 67% of the median, and the upper threshold at 133%. These values were chosen after testing with different threshold settings. These threshold settings were applied consistently throughout the process. In the experiment, the same light condition was maintained, and the threshold settings were optimised for this environment with a consistent light supply. However, variations in lighting conditions, such as dim or uneven lighting, could potentially impact the system’s performance, particularly during edge detection and contour extraction. The edge data were then used for contour detection, followed by fitting ellipses to the detected contours.

As more than one ellipse may be detected, a filtering method was proposed to select the most suitable ellipse representing the bolt or nut’s circular centre. The filtering process is illustrated in Fig. 2. The input for this step consists of the list of detected ellipses and the boundary box estimated by the YOLOv8 model. Each ellipse is evaluated individually. The first filter eliminates any ellipse larger than the boundary box, as the circular shape of the bolt or nut is assumed to be contained within the boundary box based on observations of the trained model.

Next, a second filtering step removes small ellipses, eliminating any with an area smaller than 8% of the boundary box area. This threshold was determined through testing under various workspace conditions. While ellipses smaller than the boundary box but larger than 8% of its area could still fit within the boundary box, any ellipse whose edges exceeded the box was filtered out. These thresholds were selected based on testing with different percentages of the area. After passing through the filters, the final ellipse representing the target object is selected. If only one ellipse remains, it is chosen as the target. If multiple ellipses meet the conditions, the distance between the centre of each ellipse and the centre of the boundary box is calculated. The ellipse closest to the boundary box centre is selected as the target. While the boundary box centre may not perfectly align with the target ellipse's exact centre, this method avoids selecting an ellipse that meets all conditions but is located near the boundary box's edge.

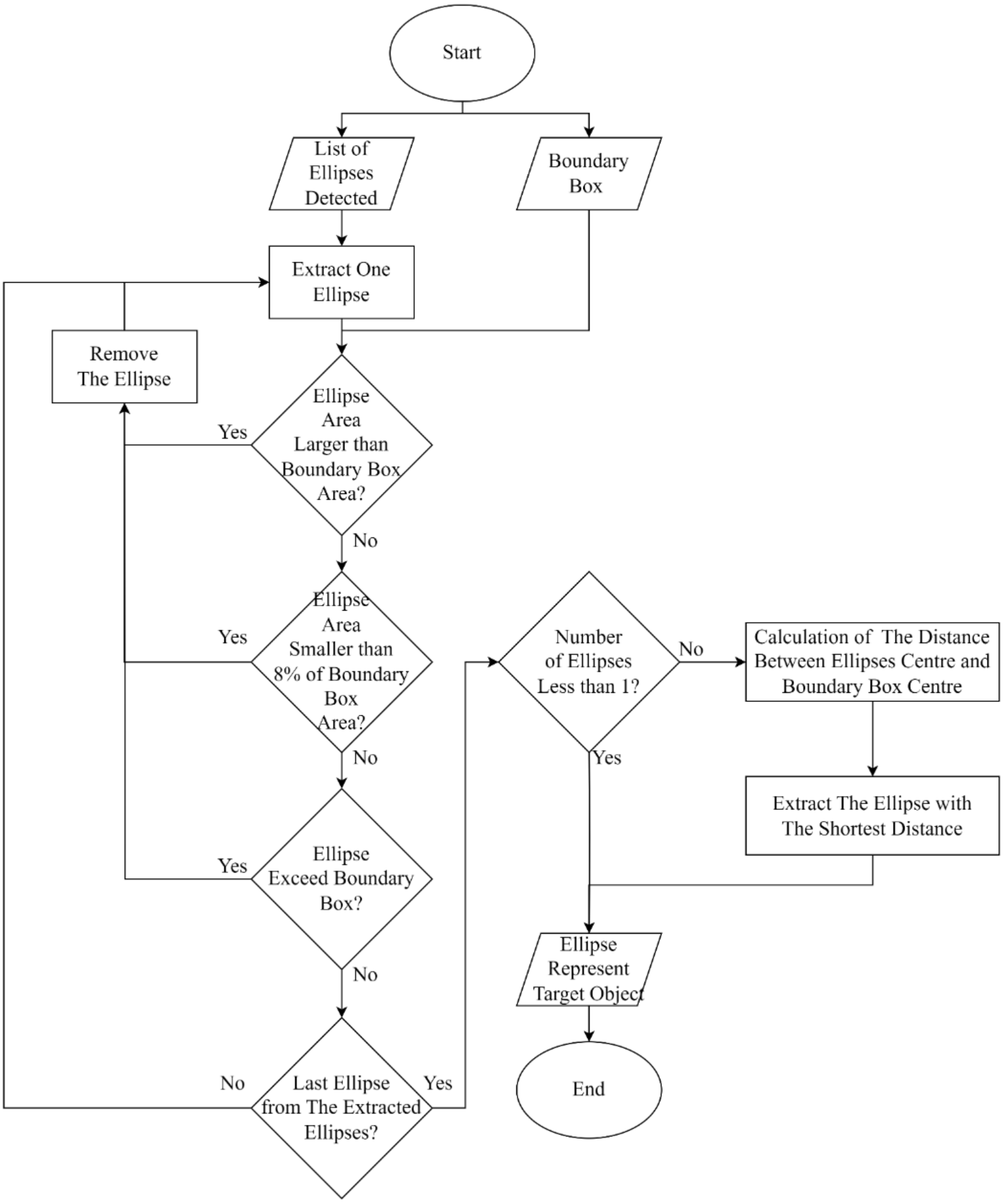

Start
List of Ellipses Detected
Boundary Box
Extract One Ellipse
Remove The Ellipse
Ellipse Area Larger than Boundary Box Area?
Yes
No
Ellipse Area Smaller than 8% of Boundary Box Area?
Yes
No
Ellipse Exceed Boundary Box?
Yes
No
Last Ellipse from The Extracted Ellipses?
No
Yes
Number of Ellipses Less than 1?
No
Yes
Calculation of The Distance Between Ellipses Centre and Boundary Box Centre
Extract The Ellipse with The Shortest Distance
Ellipse Represent Target Object
End

Fig. 2. Flowchart depicting the process of identifying the ellipse that best represents the target object from a list of detected ellipses

2.1.2. 3D Position Calculation

After selecting the ellipse to represent the bolt point or nut, the 2D position is determined. However, to perform the pick-and-place task, the 2D position alone is insufficient unless the size of the bolt and nut, along with the table height, is predefined. Therefore, the Intel RealSense D405 depth camera was used to predict the distance between the camera and the target object. Based on the position of the ellipse centre, the depth data at that position was extracted from the depth image. Using (1) and (2), the 3D position of the ellipse in the X and Y axes was calculated, while the depth data provided the Z-axis position in 3D. The X, Y, and Z axis values will be used to represent the translation component in the transformation matrix.

It is important to note that the calculated position of the ellipse is relative to the camera frame, meaning it represents the distance between the camera frame and the detected ellipse centre. In this paper, the camera was mounted on the robotic arm, following the eye-in-hand configuration. This setup was chosen because depth data collected from greater distances tends to be less accurate. By using the eye-in-hand configuration, the distance between the object and the camera can be controlled, ensuring accurate depth measurements.

Since the camera's position is not fixed, the distance between the camera and the target object alone is not sufficient for the pick-and-place process. Thus, the 3D positions of the bolt and nut were calculated relative to the robotic arm base, facilitating precise control of the robotic arm during the manipulation process. To achieve this, the forward kinematic model of the robotic arm was developed. The robotic arm used in this study was the KUKA KR4 R600, a six-degree-of-freedom manipulator.

According to the KUKA controller results, the developed forward kinematic model achieved an average difference of 1.311 × 10-5 mm in position and 0.367 × 10-5 degrees in orientation, corresponding to a percentage difference of approximately 0.2 × 10-5%. This small difference validates the accuracy of the forward kinematic model. Additionally, an extrinsic calibration method was performed to determine the distance relationship between the camera frame and the robotic arm's null tool frame. The null tool frame refers to the initial or default position of the robotic arm without any external tool attached. While the movement of the robotic arm may introduce small errors into the kinematic model, these errors are negligible and do not significantly impact the system's overall accuracy. By knowing all the transformation matrices from the robotic arm base to the target object, the 3D position of the target object relative to the robotic arm base can be accurately calculated, enabling effective pick-and-place operations.

$$X = \frac{(x - c_x) \times d}{f_x} \tag{1}$$

$$Y = \frac{(y - c_y) \times d}{f_y} \tag{2}$$

Where

X	=	X-axis position of 3D point.
Y	=	Y-axis position of 3D point.
d	=	Depth data from the depth camera.
x	=	X-axis position of 2D point.
y	=	Y-axis position of 2D point.
c_x	=	X-axis principal point of camera.
c_y	=	Y-axis principal point of camera.
f_x	=	Focal length of the camera in X-axis.
f_y	=	Focal length of the camera in Y-axis.

2.2. Optimal Capturing Positioning

In the proposed method, the camera was controlled to move to two key positions. The first position was used to capture the entire workspace, allowing the detection of the presence of the bolt and nut. The second position involved moving the camera closer to the target object to capture detailed features for more accurate calculation of its 3D position. However, testing revealed that the depth data predicted by the depth camera was inaccurate when the object was placed directly under the camera's principal point.

To address this issue, an experiment was conducted, and it was found that capturing the bolt at a distance of 3 cm away from the principal point, at an angle of 135°, provided the most suitable position for obtaining a close-up image. This is because the distance difference between the actual and calculated positions achieved the least value at this position. This positioning reduced light blockage and improved the accuracy of the depth data. It is important to note that the light source was positioned on top of the robotic arm. However, for the nut, the ideal capture position was directly above, with the camera's principal point closely aligned to the centre of the nut. This is because the depth data for the nut typically reflects the surface of the workspace table due to the hollow centre of the nut. Capturing the nut from any offset position would obstruct the camera's view, leading to less accurate results.

2.3. Experiment Setup

The experiment was divided into two parts: the first involved picking the nut from the workspace, and the second focused on placing the nut onto the bolts. In the nut-picking experiment, nuts of varying sizes from M6 to M18 were placed on the workspace and gripped using the proposed method to evaluate its accuracy. Each nut was positioned in twenty different locations on the workspace, as illustrated in Fig. 3. The chessboard used for the eye-to-hand camera calibration was also employed in this accuracy evaluation to ensure consistent placement of bolts and nuts and allow for a fair comparison across the different nut sizes. Additionally, a light mounted on top of the robotic arm provided consistent illumination during the evaluation to minimise the impact of lighting variations on the results. In the picking experiment, the success rate was calculated as the ratio of successful nut picks to the total number of trials. Similarly, in the placement experiment, the success rate was determined by the number of nuts successfully fitted onto the bolts relative to the total number of trials.

It is important to note that all nuts used in the gripping experiment were actual threaded nuts, as these were used during the training of the object detector. However, the nut placement experiment utilised threadless nuts. This decision was made because the focus of this paper is on detecting the position of the bolt and nut, and due to the limitations of the robotic arm, which is unable to perform fastening actions. Instead, threadless nuts with varying diameters, as outlined in Table 1, were fabricated for the placement experiment.

For example, an M6 nut typically has an inner diameter of 6mm, but a base diameter of 6.5mm was used in the 0% enlargement category, allowing for a 0.5mm tolerance. Enlargement percentages were calculated based on the original metric size of each nut. For instance, 5% of the original 6mm diameter of an M6 nut equates to 0.03mm, which was added to the base diameter, resulting in a final diameter of 6.8mm for an M6 nut with 5% enlargement. Each metric nut was produced with seven different levels of enlargement, with a maximum enlargement of 30%. For each enlargement size, the placement and fitting of the nut onto the bolt were tested in the same twenty positions used in the nut gripping experiment to assess the accuracy of the proposed method.

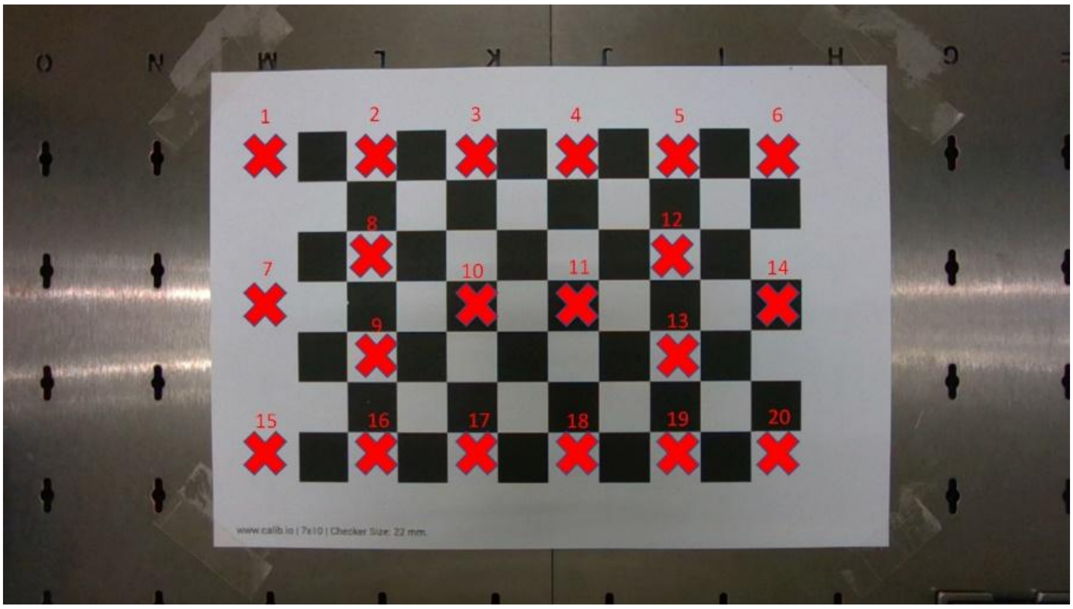


Fig. 3. Positioning of the bolt and nut for accuracy evaluation

Table 1. Diameter measurements of nuts without threading

Percentage of Enlargement (%)	Diameter of Nut Without Thread (mm)						
	M6	M8	M10	M12	M14	M16	M18
0	6.5	8.5	10.5	12.5	14.5	16.5	18.5
5	6.8	8.9	11.0	13.1	15.2	17.3	19.4
10	7.1	9.3	11.5	13.7	15.9	18.1	20.3
15	7.4	9.7	12.0	14.3	16.6	18.9	21.2
20	7.7	10.1	12.5	14.9	17.3	19.7	22.1
25	8.0	10.5	13.0	15.5	18.0	20.5	23.0
30	8.3	10.9	13.5	16.1	18.7	21.3	23.9

3. Results and Discussion

3.1. Nut Picking Performance

The nut-picking experiment revealed three distinct types of outcomes. The first condition occurred when the nuts were successfully detected by the proposed method, and the robot gripper was able to grip the nut with the correct orientation. This outcome was classified as a “success with correct orientation.” A key factor contributing to this success was the special gripper attachment, which was designed with the same edge angle as the nut, ensuring the nut was aligned into the desired orientation. The second condition arose when the nuts were gripped, but not in the correct orientation. Although the gripper managed to pick up the nut, it failed to achieve the proper alignment, leading to an outcome categorized as a “success with incorrect orientation.” Lastly, the third condition involved instances where the gripper failed to pick up any nut from the workspace. This was categorized as a “failure to grip.” These three conditions provided insight into the performance and limitations of the proposed method during the nut-picking process.

Fig. 4 presents the recorded results of the nut-picking experiments. The success rate for picking nuts in the correct orientation exceeded 80% for all nut sizes except M6. This indicates that the proposed method is effective in detecting and gripping most nuts. On average, the success rate for correctly oriented nut-picking was approximately 87%, while the success rate for incorrectly oriented nut-picking remained below 10%. Although the gripper attachment was designed to force the nut into the correct orientation, this did not occur as expected in some cases. One possible explanation is that the gripper attachments were fabricated using 3D printing technology, resulting in surfaces that were not perfectly smooth. The rough texture of the printed surfaces increased friction between the nuts and the gripper, sometimes preventing the nuts from being correctly aligned during the gripping process. Despite this, all nuts gripped with incorrect orientation were still successfully picked up.

Failures occurred only with M6 and M8 nuts, with M6 nuts showing the highest failure rate at 25%. This higher failure rate is attributed to the shorter height of M6 nuts, which makes them more sensitive to discrepancies in height calculation during the picking process. In contrast, larger nuts, such as M8 and above, have greater height, reducing their susceptibility to such discrepancies and improving their gripping success rate.

The height of the nut is determined using depth data from the camera, which can be affected by factors such as lighting conditions and the texture of the objects. As a result, the calculated height may differ slightly from the actual height. The proposed method calculates the gripping height, which is the distance between the workspace and the gripper, by positioning the gripper 2 mm above the centre of the nut. Since the centre of the nut is hollow, this height is measured from the workspace table, and the 2 mm is added based on this reference point. For the M6 nut, which is shorter, even a minor error in this height calculation can prevent the gripper from successfully gripping the nut. This setting may result in the failure to grip shorter-height nuts. One possible solution is to reduce the gripping height to 1 mm or lower. However, this increases the risk of collision between the gripper and the workspace.

Observations showed that the gripper consistently moved to the correct position above the nuts in all cases, demonstrating that the proposed method could accurately detect the nut and control the robotic arm to position it properly. However, errors in depth estimation, particularly influenced by the small inner diameter of M6 nuts, led to gripping failures. When the surface area is small, the depth camera struggles to estimate accurate depth data due to the limited information it receives from the image, affecting the gripping success.

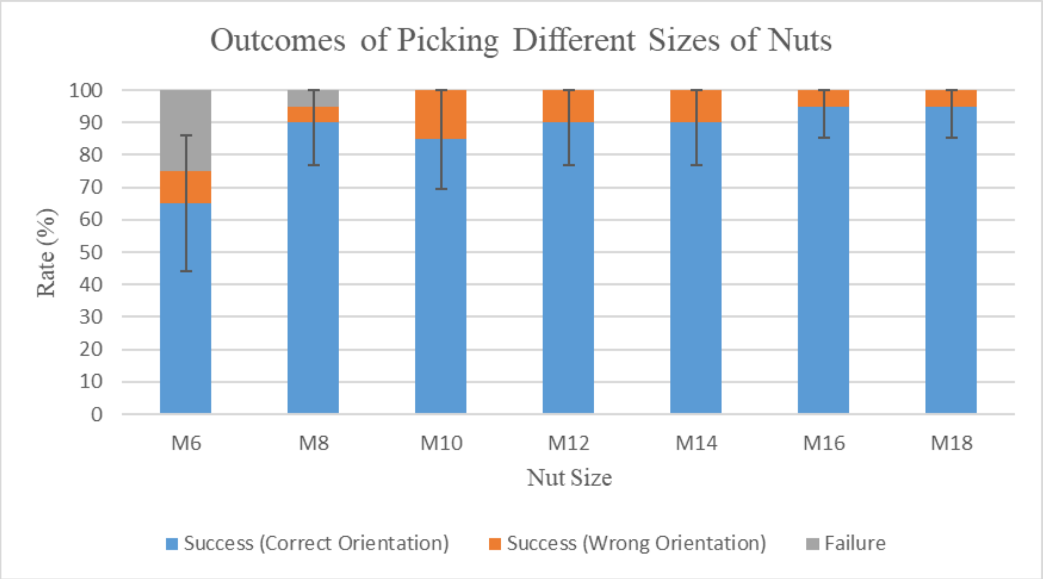


Fig. 4. Recorded outcomes of picking nuts of various sizes under controlled lighting conditions. The graph shows success rates for correct orientation (blue), success rates for wrong orientation (orange), and failure rates (grey)

During the experiment, there was one condition in which no ellipse was detected by the proposed method. When this occurred, the process was halted and restarted. This issue was observed only twice, both times with M14 nuts. These instances occurred when the camera was positioned at the global capturing location, which is farther from the nuts. Additionally, background noise near the target object contributed to the failure to detect ellipses. For example, detected ellipses may have been filtered out due to the influence of noise and other edges present in the images. This filtering issue arises when noise and distracting elements affect the detection of ellipses, leading to their exclusion by the proposed method.

Despite these occasional challenges, the percentage of instances where no ellipse was detected during nut-picking was less than 1%, which is not significant for this method. Furthermore, all nuts were successfully detected by the YOLOv8n models that were trained for this experiment. In conclusion, the proposed method successfully detects and grips nuts of all sizes, from M6 to M18, with no significant detection errors.

3.2. Nut Placement Performance

During the nut-placing experiment, four distinct conditions were observed. The first was the success condition, where the nut was successfully placed onto the bolt without making any contact with it. The second was the stuck condition, in which the nut became stuck at the bolt point and failed to fit onto the bolt. The third condition was touch-and-fit, where the nut fit onto the bolt after being released, but made contact with the bolt before the gripper released it, potentially causing slight movement or tilting of the bolt. Lastly, the failure condition occurred when the nut was dropped and did not fit onto the bolt. As shown in Fig. 5, the results of the nut-placing experiments were recorded for threadless nuts ranging from M6 to M18. Additionally, Fig. 6 illustrates the success rates for placing nuts on bolts without enlargement.

The results indicate that larger bolt sizes generally corresponded to higher success rates, and larger enlargement percentages of threadless nuts also led to increased success rates. However, some anomalies were observed where larger bolt sizes or greater nut enlargement diameters resulted in lower success rates, as shown in Table 2. For example, the success rate for 0% enlargement threadless nuts increased when placed on M6 to M12 bolts, but dropped to 60% for M14 bolts before rising again to 90% for M18 bolts.

The primary factor behind this irregularity is related to the manufacturing process of the bolts. Despite being made from the same material, the surface of the bolt points was not uniformly smooth and flat across all sizes. This inconsistency affected light reflection when the bolts were exposed to a light source, introducing noise that impacted the accuracy of edge and ellipse detection by the proposed method. As a result, this noise caused errors in calculating the 3D position of the bolts, which explains the fluctuating success rates across different bolt sizes.

Additionally, some larger enlargement threadless nuts resulted in lower success rates on the same bolt type. For instance, the 10% enlargement nuts had a lower success rate on the M8 bolt compared to the 5% enlargement nuts. Theoretically, smaller enlargement nuts should have lower success rates compared to larger ones. One key reason for this anomaly is the placement position of the bolt during the experiment. Although a chessboard was used to guide bolt placement, the bolt was not positioned exactly the same for every enlargement nut size. Variations in bolt orientation and position altered the bolt’s appearance in the images, introducing noise during edge and ellipse detection, and causing errors in calculating the bolt's 3D position. Therefore, applying machine learning algorithms with image processing methods could help improve detection accuracy in future studies by mitigating the effects of noise and positional variations.

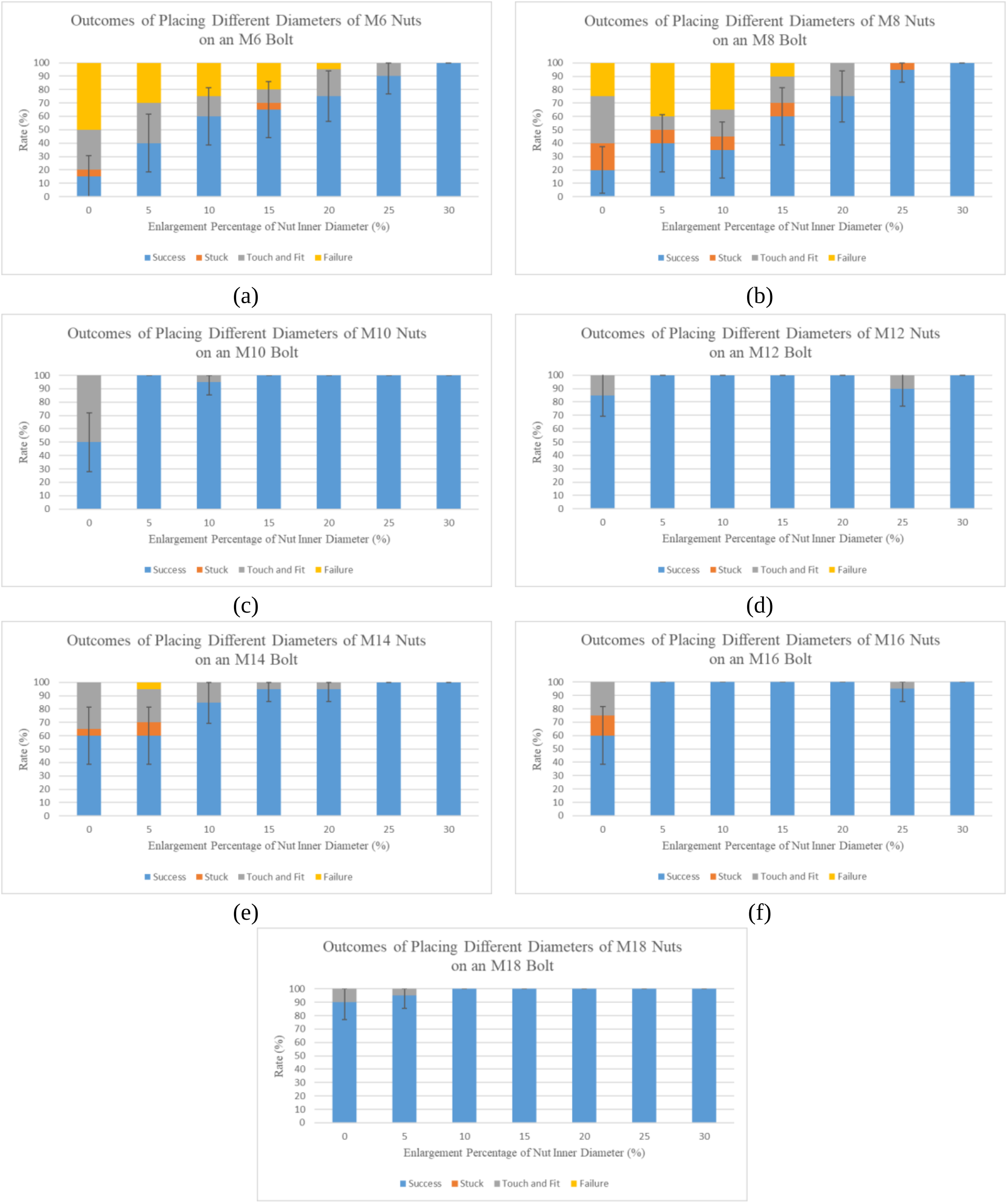
However, the differences in success rates were typically within 10% in cases where smaller enlargement nuts had higher success rates than larger ones. Since these differences are relatively minor, they do not significantly impact the overall success rate of the proposed method. On average, the success rate increased from 0% to 30% enlargement nuts. According to the results, the proposed method achieved an average success rate of 80% when 10% enlargement nuts were used. This suggests that the proposed method is effective in nut-to-bolt placement applications with 10% enlargement nuts. However, the method is less suitable for M6 and M8 bolt sizes due to the smaller size of these bolts. In future studies, a force-torque sensor could be integrated into the system to utilise the force and torque data between the bolt and nut during contact. This would enable accurate fitting of the nut without requiring any enlargement, thereby reducing dependency on labour.

While the proposed method demonstrates high success rates for larger nuts, it is important to acknowledge some limitations. The computational complexity of the method, particularly due to the integration of YOLOv8 object detection and ellipse fitting, can lead to slower processing times in real-time applications. This could impact the efficiency of the system in high-speed industrial environments where rapid picking and placing are essential. Additionally, the method's performance is highly dependent on the consistency of lighting and precise calibration, which may require additional setup and maintenance effort.

Table 2. Success rate outcomes of placing different threadless nuts onto bolts, as recorded from the experiments

Enlargement Percentage of Nut (%)	**Success Rate (%)**							
	Placing Nuts on Bolts							**Average**
	M6	**M8**	**M10**	**M12**	**M14**	**M16**	**M18**	
0	15	20	50	85	60	60	90	54.286
5	40	40	100	100	60	100	95	76.429
10	60	35	95	100	85	100	100	82.143
15	65	60	100	100	95	100	100	88.571
20	75	75	100	100	95	100	100	92.143
25	90	95	100	90	100	95	100	95.714
30	100	100	100	100	100	100	100	100.000

During the experiment of placing nuts onto bolts, there were 35 instances where no ellipses were detected and 5 instances where no bolts were detected, as detailed in Table 3. The majority of these non-detections occurred with M6 and M8 bolts, likely due to their smaller size compared to other bolt types. Notably, the number of non-ellipse detections was higher than non-bolt detections, with the latter occurring only with M6 and M8 bolts.



(a) (b)

(c) (d)

(e) (f)

(g)

Fig. 5. Recorded outcomes of placing threadless nuts, enlarged by different percentages, onto bolts of various sizes under controlled lighting conditions. The graph shows success rates (blue), stuck rates (orange), touch-and-fit rates (grey), and failure rates (yellow). Subfigures correspond to: (a) M6 bolt and nut, (b) M8 bolt and nut, (c) M10 bolt and nut, (d) M12 bolt and nut, (e) M14 bolt and nut, (f) M16 bolt and nut, and (g) M18 bolt and nut

As previously discussed, ellipse detection relies on a combination of edge detection, ellipse identification, and ellipse filtering, while bolt detection is dependent on the performance of the trained object detector. The results indicate that the YOLOv8n model successfully detected all bolts from M10 to M18, as well as most M6 and M8 bolts. Additionally, the proposed method demonstrated strong potential in detecting the majority of the elliptical shapes of the bolts. Although there were some instances of non-ellipse detection, the majority of ellipses were successfully identified, enabling accurate 3D position calculations as reflected in the experiment results.

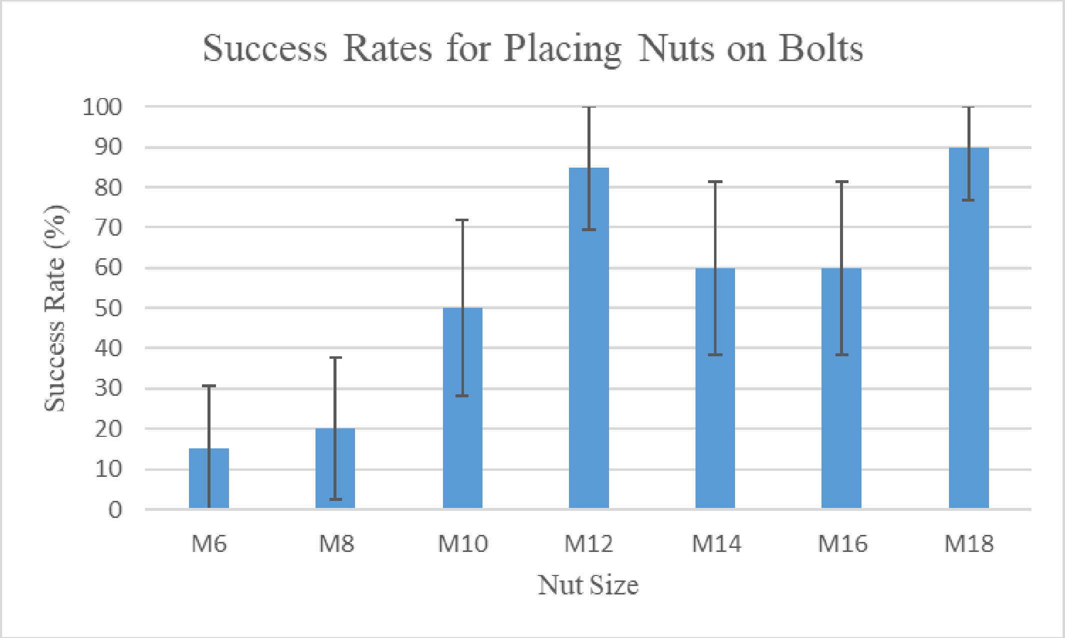


Fig. 6. Success rates for placing nuts on bolts without enlargement, showing variations across different nut sizes

One reason for the non-detection of ellipses is the smaller size of the bolts, which poses challenges for edge detection due to the limited number of pixels covering the small bolt point area. Furthermore, the non-flat and rough surface of the bolt point can introduce noise during the edge and ellipse detection process. Noise can result in false edges, which may mislead contour and ellipse detection, causing the detected position of the bolt to be inaccurate. Despite these challenges, the overall performance of the proposed method remained robust, with instances of both non-ellipse and non-bolt detections accounting for less than 1% of the total experiments. This low rate of non-detections underscores the effectiveness of the proposed method in accurately detecting and placing nuts onto bolts across a range of sizes. To mitigate the non-detection of bolts and ellipses, the training dataset can be increased, and the threshold values of the Canny edge detector can be adjusted.

Table 3. Total number of non-ellipse and non-bolt detections during the accuracy experiment for placing nuts onto bolts

	Placing Nuts on Bolts							**Total**
	M6	**M8**	**M10**	**M12**	**M14**	**M16**	**M18**	
Number of Non-Ellipses Detected	10	12	1	6	2	2	2	35
Number of Non-Bolts Detected	4	1	0	0	0	0	0	5

4. Conclusion

This research successfully developed an intelligent control system for a vision-equipped robotic arm capable of autonomously and accurately picking and placing nuts onto randomly arranged bolts, especially for larger sizes of bolts and nuts, with potential applications in industrial automation and robotics. The system operates without requiring predefined positions or markers, enhancing the flexibility and efficiency of industrial assembly lines and reducing dependency on manual labour. The system integrates object detection with edge and ellipse detection to precisely locate bolt and nut centres. The method achieved an 87% success rate for picking nuts with correct orientation. Specifically, the success rates were 65% for M6 nuts, 85% for M10 nuts, and over 90% for M8 and larger nuts. Failures were primarily observed for M6 and M8 nuts due to their shorter height, which increased sensitivity to depth estimation errors. The object detector identified all nuts, with only two instances of ellipse detection failure. For nut placement, a 90% success rate was achieved by enlarging nuts by 25% for M6, M8, and M14 bolts, and by 5% for M10, M12, and M16 bolts. M18 bolts required no enlargement. This enlargement was necessary to increase the likelihood of accurate fitting, as it helped address issues related to tolerances and mechanical alignment between the bolt and nut. Factors like bolt surface conditions, image noise, and depth estimation errors influenced the results. Overall, non-detections were less than 1%, demonstrating the system's effectiveness in accurately performing the task. However, it requires longer processing times and a high-performance computer. This study assumes that the bolt and nut are placed with their circular surfaces parallel to the workspace and camera frame, which simplifies detection but may limit performance in real-world applications where such alignment cannot be guaranteed. Furthermore, this method has the potential to adapt to other assembly applications involving components with elliptical or circular features. Examples include the assembly of gears, bearings, or circular mechanical parts in various industrial settings, highlighting its versatility.

Several recommendations for future work arise from this project’s findings. While the method successfully handled M10 to M18 threadless nuts due to weight limitations of the robotic arm’s end effector, future research could explore equipping the arm with a tightening tool or incorporating torque application to work with standard, threaded nuts, broadening its applicability. Another suggestion is integrating an eye-to-hand camera for obstacle detection and collision-free operations. Additionally, incorporating machine learning algorithms could enhance the system’s ability to adapt to varying nut and bolt sizes, while adaptive control techniques might improve real-time alignment during the bolt-nut mating process. Adding a force sensor for improved alignment in the bolt-nut mating process could also enhance assembly reliability. Additionally, improving depth estimation technology for smaller objects could expand the system’s capability to handle a wider range of bolt and nut sizes. Furthermore, future studies could explore integrating multiple robots to improve efficiency in complex tasks. For example, one robot could perform detection tasks while another handles manipulation, streamlining the assembly process and reducing overall operation time. A sensitivity analysis could be conducted in future work to evaluate the robustness of the system under varying lighting conditions and to determine the optimal thresholds for different environments. This would provide insights into the adaptability and reliability of the proposed method in diverse real-world settings.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by TAR UMT internal grant, project number UC/I/G2022-00094 and 2024 GISU Joint Research Project.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] J. Xu, C. Zhang, Z. Liu and Y. Pei, “A Review on Significant Technologies Related to the Robot-Guided Intelligent Bolt Assembly Under Complex or Uncertain Working Conditions,” *IEEE Access*, vol. 7, pp. 136752-136776, 2019, https://doi.org/10.1109/ACCESS.2019.2941918.

[2] J. M. Tao, J. Y. S. Luh and Y. F. Zheng, “Compliant coordination control of two moving industrial robots,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 322-330, 1990, https://doi.org/10.1109/70.56664.

[3] B. Chu, K. Jung, K. H. Ko and D. Hong, “Mechanism and analysis of a robotic bolting device for steel beam assembly,” *ICCAS 2010*, pp. 2351-2356, 2010, https://doi.org/10.1109/ICCAS.2010.5669929.

[4] F. Zhang, L. Hua, Y. Fu and B. Guo, “Dynamic simulation and analysis for bolt and nut mating of dual arm robot,” *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 660-665, 2012, https://doi.org/10.1109/ROBIO.2012.6491042.

[5] K. Inamura, K. Iwasaki, R. Kunimura, S. Hamasaki and H. Osumi, “Investigation of Screw Fastening by Human-Robot Cooperation,” *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1-6, 2023, https://doi.org/10.1109/SII55687.2023.10039285.

[6] M. Liu *et al*., “A robotic Hi-Lite Bolts/collars assembly system and control strategy,” *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2255-2260, 2017, https://doi.org/10.1109/ROBIO.2017.8324754.

[7] K. Pfeiffer, A. Escande and A. Kheddar, “Nut fastening with a humanoid robot,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6142-6148, 2017, https://doi.org/10.1109/IROS.2017.8206515.

[8] B. H. Yoshimi and P. K. Allen, “Integrating real-time vision and manipulation,” *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, vol. 5, pp. 178-187, 1997, https://doi.org/10.1109/HICSS.1997.663173.

[9] M. A. Elaziz *et al*., “An Improved Marine Predators Algorithm With Fuzzy Entropy for Multi-Level Thresholding: Real World Example of COVID-19 CT Image Segmentation,” *IEEE Access*, vol. 8, pp. 125306-125330, 2020, https://doi.org/10.1109/ACCESS.2020.3007928.

[10] M. Wang, Z. Zhang, X. Qiu, S. Gao and Y. Wang, “ATASI-Net: An Efficient Sparse Reconstruction Network for Tomographic SAR Imaging With Adaptive Threshold,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-18, 2023, https://doi.org/10.1109/TGRS.2023.3268132.

[11] W. Zhu, H. Gu, Z. Fan and X. Zhu, “Robust Stereo Road Image Segmentation Using Threshold Selection Optimization Method Based on Persistent Homology,” *IEEE Access*, vol. 11, pp. 122221-122230, 2023, https://doi.org/10.1109/ACCESS.2023.3329056.

[12] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979, https://doi.org/10.1109/TSMC.1979.4310076.

[13] J. Yen, F. Chang and S. Chang, “A new criterion for automatic multilevel thresholding,” *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370-378, 1995, https://doi.org/10.1109/83.366472.

[14] K. Chen, Y. Zhou, Z. Zhang, M. Dai, Y. Chao, and J. Shi, “Multilevel Image Segmentation Based on an Improved Firefly Algorithm,” *Mathematical Problems in Engineering*, vol. 2016, no. 1, pp. 1-12, 2016, https://doi.org/10.1155/2016/1578056.

[15] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, “Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation,” *Expert Systems with Applications*, vol. 83, pp. 242–256, 2017, https://doi.org/10.1016/j.eswa.2017.04.023.

[16] P. Prathusha and S. Jyothi, “A Novel Edge Detection Algorithm for Fast and Efficient Image Segmentation,” *Data Engineering and Intelligent Computing*, pp. 283–291, 2018, https://doi.org/10.1007/978-981-10-3223-3_26.

[17] Y. Chong, X. Chen and S. Pan, “Context Union Edge Network for Semantic Segmentation of Small-Scale Objects in Very High Resolution Remote Sensing Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, https://doi.org/10.1109/LGRS.2020.3021210.

[18] J. Li, F. Pu, H. Chen, X. Xu and Y. Yu, “Crop Segmentation of Unmanned Aerial Vehicle Imagery Using Edge Enhancement Network,” *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1-5, 2024, https://doi.org/10.1109/LGRS.2024.3358983.

[19] C. Wang, H. Chen and S. Zhao, “RERN: Rich Edge Features Refinement Detection Network for Polycrystalline Solar Cell Defect Segmentation,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1408-1419, 2024, https://doi.org/10.1109/TII.2023.3275705.

[20] F. He, M. A. Parvez Mahmud, A. Z. Kouzani, A. Anwar, F. Jiang, and S. H. Ling, “An Improved SLIC Algorithm for Segmentation of Microscopic Cell Images,” *Biomedical Signal Processing and Control*, vol. 73, p. 103464, 2022, https://doi.org/10.1016/j.bspc.2021.103464.

[21] M. S. Chaibou, K. Kalti, B. Solaiman and M. A. Mahjoub, "A Combined Approach Based on Fuzzy Classification and Contextual Region Growing to Image Segmentation," *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pp. 172-177, 2016, https://doi.org/10.1109/CGiV.2016.41.

[22] E. O. Rodrigues, A. Conci and P. Liatsis, "ELEMENT: Multi-Modal Retinal Vessel Segmentation Based on a Coupled Region Growing and Machine Learning Approach," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3507-3519, 2020, https://doi.org/10.1109/JBHI.2020.2999257.

[23] X. Wang and S. Ji, "Roof Plane Segmentation From LiDAR Point Cloud Data Using Region Expansion Based L0 Gradient Minimization and Graph Cut," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10101-10116, 2021, https://doi.org/10.1109/JSTARS.2021.3113083.

[24] Y. Li, Z. Li, Z. Guo, A. Siddique, Y. Liu and K. Yu, "Infrared Small Target Detection Based on Adaptive Region Growing Algorithm With Iterative Threshold Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-15, 2024, https://doi.org/10.1109/TGRS.2024.3376425.

[25] C. Senthilkumar and R. K. Gnanamurthy, "A Fuzzy clustering based MRI brain image segmentation using back propagation neural networks," *Cluster Computing*, vol. 22, pp. 12305–12312, 2018, https://doi.org/10.1007/s10586-017-1613-x.

[26] C. Hung, J. Underwood, J. Nieto, and S. Sukkarieh, "A Feature Learning Based Approach for Automated Fruit Yield Estimation," *Field and Service Robotics*, pp. 485–498, 2020, https://doi.org/10.1007/978-3-319-07488-7_33.

[27] H. Chen *et al*., "Unsupervised Local Discrimination for Medical Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15912-15929, 2023, https://doi.org/10.1109/TPAMI.2023.3299038.

[28] Y. Chen, Z. Wang and X. Bai, "Fuzzy Sparse Subspace Clustering for Infrared Image Segmentation," *IEEE Transactions on Image Processing*, vol. 32, pp. 2132-2146, 2023, https://doi.org/10.1109/TIP.2023.3263102.

[29] X. Tian, T. Sun and Y. Qi, "Ancient Chinese Character Image Segmentation Based on Interval-Valued Hesitant Fuzzy Set," *IEEE Access*, vol. 8, pp. 146577-146587, 2020, https://doi.org/10.1109/ACCESS.2020.3014219.

[30] J. Fu, J. Zhao and F. Li, "Infrared Sea-Sky Line Detection Utilizing Self-Adaptive Laplacian of Gaussian Filter and Visual-Saliency-Based Probabilistic Hough Transform," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, https://doi.org/10.1109/LGRS.2021.3111099.

[31] Y. Ding, Y. Sun, X. Yu, D. Cheng, X. Lin and X. Xu, "Bezier-Based Hough Transforms for Doppler Localization of Human Targets," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 1, pp. 173-177, 2020, https://doi.org/10.1109/LAWP.2019.2956842.

[32] Z. Long *et al*., "Motor Fault Diagnosis Based on Scale Invariant Image Features," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1605-1617, 2022, https://doi.org/10.1109/TII.2021.3084615.

[33] G. Flitton, T. P. Breckon, and N. Megherbi, "Object recognition using 3D SIFT in complex CT volumes," *Applied Mathematics and Computing Group School of Engineering*, pp. 1–12, 2010, https://bmva-archive.org.uk/bmvc/2010/conference/paper11/abstract11.pdf.

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision – ECCV 2006*, vol. 110, no. 3, pp. 346–359, 2008, https://doi.org/10.1007/11744023_32.

[35] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011, https://doi.org/10.1109/ICCV.2011.6126544.

[36] G. Dong and G. Kuang, "Classification on the Monogenic Scale Space: Application to Target Recognition in SAR Image," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2527-2539, 2015, https://doi.org/10.1109/TIP.2015.2421440.

[37] A. Novak, N. Armstrong, T. Caelli and I. Blair, "Bayesian Contrast Measures and Clutter Distribution Determinants of Human Target Detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp.

1115-1126, 2017, https://doi.org/10.1109/TIP.2016.2644269.

[38] K. Vodrahalli and A. K. Bhowmik, "3D computer vision based on machine learning with deep neural networks: A review," *Journal of the Society for Information Display*, vol. 25, no. 11, pp. 676–694, 2017, https://doi.org/10.1002/jsid.617.

[39] F. Gao, T. Huang, J. Sun, J. Wang, A. Hussain, and E. Yang, "A New Algorithm for SAR Image Target Recognition Based on an Improved Deep Convolutional Neural Network," *Cognitive Computation*, vol. 11, no. 6, pp. 809–824, 2019, https://doi.org/10.1007/s12559-018-9563-z.

[40] Y. Tian, H. Meng and F. Yuan, "Multiscale and Multilevel Enhanced Features for Ship Target Recognition in Complex Environments," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 4640-4650, 2024, https://doi.org/10.1109/TII.2023.3327570.

[41] M. Gong and Y. Shu, "Real-Time Detection and Motion Recognition of Human Moving Objects Based on Deep Learning and Multi-Scale Feature Fusion in Video," *IEEE Access*, vol. 8, pp. 25811-25822, 2020, https://doi.org/10.1109/ACCESS.2020.2971283.

[42] F. Yu, B. He and J. -X. Liu, "Underwater Targets Recognition Based on Multiple AUVs Cooperative via Recurrent Transfer-Adaptive Learning (RTAL)," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1574-1585, 2023, https://doi.org/10.1109/TVT.2022.3211862.

[43] K. Tseng, Y. Zhang, Q. Zhu, K. L. Yung, and W. H. Ip, "Semi-supervised image depth prediction with deep learning and binocular algorithms," *Applied Soft Computing*, vol. 92, p. 106272, 2020, https://doi.org/10.1016/j.asoc.2020.106272.

[44] F. Deng, L. Zhang, F. Gao, H. Qiu, X. Gao and J. Chen, "Long-Range Binocular Vision Target Geolocation Using Handheld Electronic Devices in Outdoor Environment," *IEEE Transactions on Image Processing*, vol. 29, pp. 5531-5541, 2020, https://doi.org/10.1109/TIP.2020.2984898.

[45] J. Zhao and R. S. Allison, "The Role of Binocular Vision in Avoiding Virtual Obstacles While Walking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3277-3288, 2021, https://doi.org/10.1109/TVCG.2020.2969181.

[46] M. Chen, W. Lu, X. Liu, S. Li, F. Gao and F. Shuang, "Hybrid Vision/Force Control of Robotic Autonomous Assembly for 110kV Hot-line Maintenance," *2021 China Automation Congress (CAC)*, pp. 690-695, 2021, https://doi.org/10.1109/CAC53003.2021.9727413.

[47] O. Salunkhe, O. Stensöta, M. Åkerman, Å. F. Berglund, and P. A. Alveflo, "Assembly 4.0: Wheel hub nut assembly using a cobot," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1632–1637, 2019, https://doi.org/10.1016/j.ifacol.2019.11.434.

[48] M. Koike, K. Kurabe, K. Yamashita, Y. Kato, K. Jinno and K. Tatsuno, "An approach to object recognition for a power distribution line maintenance robot. The case of identifying a mechanical bolt to be tightened with a nut," *2016 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, pp. 1-7, 2016, https://doi.org/10.1109/MHS.2016.7824219.

[49] Y. Chen, J. Yu, L. Shen, Z. Lin and Z. Liu, "Vision-Based High-Precision Assembly with Force Feedback," *2023 9th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 399-404, 2023, https://doi.org/10.1109/ICCAR57134.2023.10151762.

[50] R. Holladay, T. Lozano-Pérez and A. Rodriguez, "Planning for Multi-stage Forceful Manipulation," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6556-6562, 2021, https://doi.org/10.1109/ICRA48506.2021.9561233.

[51] D. Son, H. Yang and D. Lee, "Sim-to-Real Transfer of Bolting Tasks with Tight Tolerance," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9056-9063, 2020, https://doi.org/10.1109/IROS45743.2020.9341644.

[52] T. S. H. Ting *et al.*, "A Review of Advanced Force Torque Control Strategies for Precise Nut-to-Bolt Mating in Robotic Assembly," *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 139-158, 2024, https://doi.org/10.31763/ijrcs.v5i1.1604.

[53] B. Shi, H. Liu, E. Zappa and X. Ni, "A Looseness Recognition Method for Rail Fastener Based on Semantic Segmentation and Fringe Projection Profilometry," *IEEE Sensors Journal*, vol. 24, no. 11, pp. 18608-18621, 2024, https://doi.org/10.1109/JSEN.2024.3389297.

[54] N. Akai, T. Hirayama and H. Murase, “Semantic Localization Considering Uncertainty of Object Recognition,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4384-4391, 2020, https://doi.org/10.1109/LRA.2020.2998403.

[55] X. Ma, Q. Wu, X. Zhao, X. Zhang, M. -O. Pun and B. Huang, “SAM-Assisted Remote Sensing Imagery Semantic Segmentation With Object and Boundary Constraints,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-16, 2024, https://doi.org/10.1109/TGRS.2024.3443420.

[56] M. I. M. Ameerdin, M. H. Jamaluddin, A. Z. Shukor, and S. Mohamad, “A Review of Deep Learning-Based Defect Detection and Panel Localization for Photovoltaic Panel Surveillance System,” *International Journal of Robotics and Control Systems*, vol. 4, no. 4, pp. 1746–1771, 2024, https://doi.org/10.31763/ijrcs.v4i4.1579.

[57] G. Chuang and C. Li-Jia, “Bridge Crack Detection Based on Attention Mechanism,” *International Journal of Robotics and Control Systems*, vol. 3, no. 2, pp. 259–269, 2023, https://doi.org/10.31763/ijrcs.v3i2.929.

[58] A. N. Handayani, F. A. Pusparani, D. Lestari, I. M. Wirawan, A. P. Wibawa, and O. Fukuda, “Real-Time Obstacle Detection for Unmanned Surface Vehicle Maneuver,” *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 765–779, 2023, https://doi.org/10.31763/ijrcs.v3i4.1147.

[59] I. G. S. M. Diyasa *et al*., “Enhanced Human Hitting Movement Recognition Using Motion History Image and Approximated Ellipse Techniques,” *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 222–239, 2024, https://doi.org/10.31763/ijrcs.v5i1.1599.

[60] X. Yue, K. Qi, X. Na, Y. Zhang, Y. Liu, C. Liu, “Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage,” *Agriculture*, vol. 13, no. 8, p. 1643, 2023, https://doi.org/10.3390/agriculture13081643.

[61] Y. K. Tan, K. M. Chin, T. S. H. Ting, Y. H. Goh and T. H. Chiew, “Research on YOLOv8 Application in Bolt and Nut Detection for Robotic Arm Vision,” *2024 16th International Conference on Knowledge and Smart Technology (KST)*, pp. 126-131, 2024, https://doi.org/10.1109/KST61284.2024.10499651.