Journal of Intelligent & Robotic Systems (2024) 110:159
https://doi.org/10.1007/s10846-024-02195-z

REGULAR PAPER

Integration of Artificial Vision and Image Processing into a Pick and Place Collaborative Robotic System

Adriano A. Santos[1,2] · Cas Schreurs[1] · António Ferreira da Silva[1,2] · Filipe Pereira[3] · Carlos Felgueiras[4] · António M. Lopes[2] · José Machado[3]

Received: 1 May 2024 / Accepted: 21 October 2024 / Published online: 8 November 2024

Abstract

In the field of robotics, pick and place applications are becoming increasingly popular due to their ability to automate repetitive tasks that can create temporary or permanent injuries. To enhance the efficiency of these applications, object recognition using a fixed camera or one mounted on a robotic hand has been employed. This paper explores the possibilities of implementing a low-cost camera into a collaborative robotic system. A software architecture has been developed, including modules for perception, pick and place, and part transfer. A comprehensive overview of various intuitive drag-and-drop image processing technologies and their suitability for object recognition in a robotic context is provided. The challenges related to lighting and the effect of shadows in object recognition are discussed. A critical assessment is made of the architecture development platform as well as the study and the results are performed, and the effectiveness of the proposed solution based on the Niop architecture is verified.

Keywords Artificial vision · Camera · Cobot · Image processing · Object detection · UR3e collaborative robot

1 Introduction

In addressing current challenges such as labor and skill shortages, collaborative robots (also known as cobots) can provide solutions. By integrating them into a workplace, repetitive tasks can be performed by cobots, while more value-added tasks can still be carried out by human workers. Additionally, cobots can ensure that human workers operating alongside them are not exposed to hazardous environments or engage in ergonomically unfavorable actions, resulting in a reduction of the biomechanical workload with a consequent improvement in the well-being of operators [1]. As a result, the use of cobots impacts not only the way we work but also the way we live.

A collaborative robot can execute various actions, but it requires well-defined conditions to operate flawlessly. For example, in a pick and place action, the robot needs to ensure that the object to be picked is in a predefined position and

✉ Adriano A. Santos
ads@isep.ipp.pt

Cas Schreurs
schreurs.cas@gmail.com

António Ferreira da Silva
afs@isep.ipp.pt

Filipe Pereira
filipe.as.pereira@gmail.com

Carlos Felgueiras
mcf@isep.ipp.pt

António M. Lopes
aml@fe.up.pt

José Machado
jmachado@dem.uminho.pt

1 CIDEM-School of Engineering (ISEP), Polytechnic of Porto (P.Porto), Rua Dr. António Bernardino de Almeida 431, 4249-015 Porto, Portugal

2 INEGI-Institute of Science and Innovation in Mechanical and Industrial Engineering, 4200-465 Porto, Portugal

3 MEtRICs-Research Centre, School of Engineering, University of Minho, Campus of Azurém, 4800-058 Guimarães, Portugal

4 CIETI-School of Engineering (ISEP), Polytechnic of Porto (P.Porto), Rua Dr. António Bernardino de Almeida 431, 4249-015, Porto, Portugal

orientation. This is where a vision system comes into play. Integrating a vision system into the robot, although being a difficult task due to the variety of products [2], enables it to perceive the orientation, position and even the color of the object [3]. Therefore, adding a vision system to a robot offers significant advantages in recognizing the surrounding environment [4].

Vision systems that are integrated into collaborative robots improve accuracy and adaptation in dynamic environments in manufacturing processes such as Pick & Place, allowing cobots to perform tasks with greater precision and adjust their operations in real time based on visual feedback that you perceive around you. Furthermore, these systems, combined with deep learning algorithms, enable cobots to recognize objects, anticipate changes in the work environment and perform complex tasks, such as assembly of small parts and quality inspection, increasing efficiency and safety in industrial processes. from Pick & Place [5].

Collaborative robots have appeared on the market in order to respond to a growing variety of industrial and service applications, with the application depending on the location of application and the type of collaboration established with humans. This evolution is driven by the integration of advanced technologies, such as artificial intelligence (AI) and machine learning systems, which substantially improve the ability of cobots to perceive and interact with the environment in which they operate, becoming very important tools in industrial environments. production and production tasks. [6].

The increasing use of artificial intelligence in human–robot collaboration demonstrates how AI algorithms can be integrated into collaborative robotics systems to improve safety, predict failures, and optimize tasks, especially in complex industrial environments. This allows collaborative robots to be more effective in their responses and capable of performing cognitive tasks more effectively [7–10].

The investigations carried out by Ali Sezer et. all about defect detection methods using deep learning models, have shown effectiveness in early identification of defects in printed circuit board production processes. This study highlights the use of convolutional neural networks (CNN) combined with optimization techniques to improve real-time defect detection, which is critical to maintaining production efficiency and quality [11].

Similarly, in the field of agriculture, İlayda Yağ et. all demonstrated the applicability of hybrid artificial intelligence algorithms for the real-time detection of plant diseases, using a combination of deep learning techniques. This hybrid model created highlights the importance of accuracy and efficiency in high-variability environments, such as agriculture, where conditions can change quickly. Furthermore, the authors state that there are systems that do not allow the integration of algorithms that can deal with chaoticity in 2D images in Pick and Place systems [12].

Taking into account the previously mentioned investigations, our study proposes a collaborative robotic system that not only recognizes and manipulates objects with high precision, but also efficiently adapts to variations in the work environment in which it performs a Pick & Place activity. Furthermore, our research differentiates itself by its ability to apply similar computer vision concepts in a collaborative robotic environment, where the integration of low-cost cameras not only improves the efficiency of Pick & Place tasks, but also makes the solution more efficient and accessible for different production scales.

The main objective of this work is the integration of artificial vision and image processing with a low-cost camera into a pick and place collaborative robotic system. Using a collaborative robot UR3e equipped with an Intel vision system, the aim is to identify randomly placed objects on a designated surface. The system must recognize the object's color, orientation, contour, etc., and place it in a vertical storage or a specific location on a worktable.

Based on the main objective of this work, several innovations and distinctive approaches stand out, significantly advancing the integration of artificial vision and image processing in collaborative pick and place robotic systems. These include:

- **Correct Integration of Lighting Source in Robotic Systems:** The paper introduces the use of a lighting source to enhance image quality and improve object recognition rates, leveraging the block programming offered by the NIOP tool.
- **Approaching Shadows in Object Detection:** This paper acknowledges and addresses the issue of shadows caused by the light source, ensuring uniform illumination to enhance the reliability of object detection results.
- **Consideration of Object Location and Rotation:** Enables methods to obtain precise object location for successful pick and place tasks, offering insights for optimization, including software adjustments relative to actual positioning.
- **Future Directions in Object Recognition with Vision Systems in Automation and Robotics:** The paper describes future advances, such as the development of adaptive algorithms, expansion of recognition regions, and integration of deep learning techniques, to improve object recognition.
- **Deep Learning in Vision:** The work shows how using deep learning in vision systems helps cobots identify and classify objects in real time, making task execution more accurate.
- **Adaptation in Dynamic Factories:** The work also shows how the low-cost vision system can help cobots quickly adjust to changes in the work environment, ensuring quality and safety in industrial settings.

- **Added Value for Smart Factories:** The work demonstrates how combining AI and vision in cobots can facilitate the automation of more complex tasks, such as quality inspection and assembly of fragile parts (e.g., electronic chips), in modern factories/industries.

The method proposed in this work offers several advantages over existing systems, including the integration of a low-cost camera into a collaborative robotic system, the improvement in image quality and object recognition, the integration of AI with image processing, lighting control with shadow elimination in object detection and efficient adaptation to dynamic environments. These advantages are explored in detail throughout the article.

For this system, the authors chose a low-cost camera due to the following reasons:

- In many industrial applications, especially small and medium-sized businesses (SMEs), budget is a critical factor.
- High-performance vision systems, although accurate, are inaccessible due to their high cost.
- The use of low-cost cameras makes the technology more accessible, enabling wider adoption of this technology at an industrial level.
- This type of cameras offers simple interfaces and APIs (Application Programming Interfaces) that allow quick integration with the robot control software.
- The use of low-cost cameras reduces maintenance costs and facilitates replacement in case of failure, minimizing downtime, which allows for an improvement in terms of maintenance.
- This type of cameras offers sufficient precision for most Pick & Place tasks, especially when combined with computer vision algorithms, as is the case with the system proposed by the authors.
- A low-cost camera is very effective in dynamic environments where objects can be in different positions and orientations.

This paper is organized into 7 Sections. Section 2 deals with the state of the art on the subject, emphasizing the main gaps in existing systems. Section 3 explains the materials and methods used in this system. Section 4 explains hardware and software utilized in the proposed solution. Section 5, describes the solution's architecture, being divided into four subparts: object detection, calibration, data communication, and pick and place operation. Potential methods for extracting position and rotational information from the image are examined. The calibration method is outlined, followed by an explanation of how parameters are transmitted from the software to the robot. Finally, an elucidation of the robot code's creation process is presented. Section 6 presents and discusses the main results. Finally, Sect. 7 outlines the conclusions.

2 State of the Art

The state of the art will delve into the world of computer vision and image manipulation. The aim of this discussion is to provide the reader with a comprehensive understanding of the fundamental concepts and features in this field, thus facilitating comprehension of the concepts presented. It will encompass various topics such as image processing techniques, computer vision algorithms, object recognition, and image segmentation. This Section will serve as the foundation for the rest of the paper.

This chapter provides a review and analysis of existing technologies for the "pick and place" task in collaborative robotics, exploring advances in computer vision, AI algorithms for trajectories and the integration of systems and sensors. According to Muhammad Umar Anjum et. All, the "pick and place" task is one of the most critical in industrial automation, especially in assembly lines and manufacturing processes where precision and efficiency are important. Also, the same authors state that collaborative robotics allows safe interaction between humans and robots, and that it is a promising solution that allows increasing productivity without compromising the safety of people and the facilities in question [13].

2.1 Computer Vision and Object Detection

Muhammad Umar Anjum et. all presents a hybrid method for object detection using a camera mounted on the wrist of a UR5 cobot in Fig. 1. The approach combines statistical techniques with error reduction methods to estimate the position of objects with high precision, achieving 99.785% accuracy in locating objects in a simulated environment [13].

The technique was applied in a fruit sorting task, demonstrating its versatility. Also was used AI to identify the objects with a YOLO algorithm in a simulation concept [13]. Figure 2 shows the simulation used, where boxes were placed to place the objects and the fruits were placed on a small table. The initial position of the cobot is defined by the user.

Based on our research, this work presents the following limitations [13]:

- Effective method only in controlled environments;
- System that will have difficulties adapting to more complex or dynamic environments, such as those in which there will be disorganized objects or with different lighting conditions.

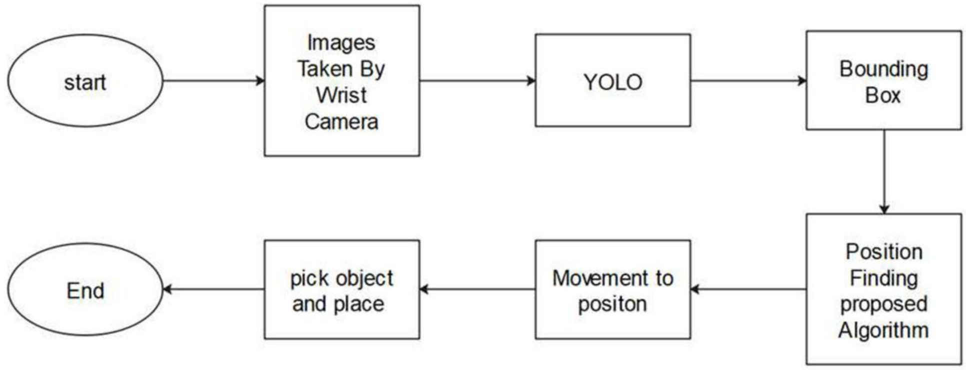


Fig. 1 Used vision system [13]

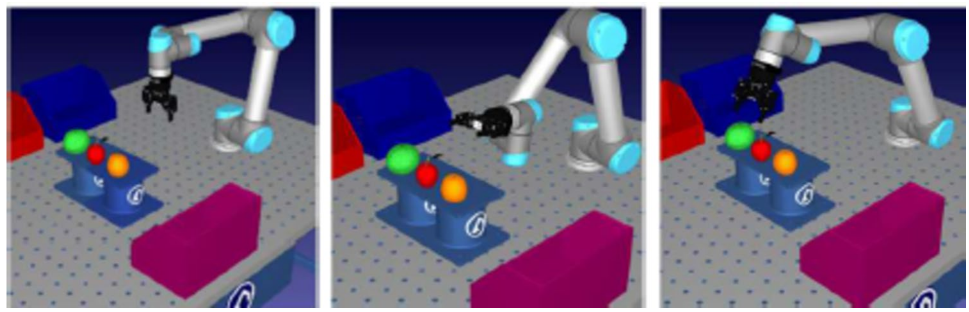


Fig. 2 Used vision system [13]

- The technique depends on pre-established correlations between image coordinates and real-world coordinates, which limits its applicability in less than perfect scenarios.
- The proposed solution is based solely on simulation and there are no results or proposals for integration in an industrial environment.
- Should a cobot with artificial vision and image processing be used in the future, it could open up many more possibilities in the future.
- Integration of algorithms with an RGBD camera and computer vision would be beneficial.
- The proposed system was designed for simpler environments. More complex environments should be tested in the future, as well as the use of scattered objects inserted in a disordered environment.

2.2 Pick and Place Sequence Optimization

Jorge Borrell et. all carried out a study to optimize "pick and place" sequences in two-arm collaborative robots represented in Fig. 3, using binary linear programming (BILP) to minimize the total operating time in a footwear production line [14].

The model proposed by the authors determines the ideal sequence of movements for the ABB cobot, optimizing efficiency and reducing downtime. Research has demonstrated a significant improvement in assembly speed in a specific industrial environment [14].

Figure 4 represents the pick-and-place task created by authors with the BILP model, has been tested in experimental environment to some manufacture in different scenarios to validate it.

Based on our research, this work presents the following limitations [14]:

- The system only focuses on trajectory optimization and distance minimization for pick-and-place operations, but does not have the correct integration of a lighting source to improve image quality and, consequently, object recognition rates.
- The system in question does not use shadow treatment caused by light sources.
- The authors addressed the location and orientation of objects, but the focus is only mainly on optimizing the trajectory and minimizing operating time, but lacks information on how the software can be adjusted to account for differences between the predicted and actual position.
- The work developed by the authors does not have adaptive algorithms and does not allow for the expansion of recognition regions or the integration of deep learning techniques.
- Although the authors use mathematical programming techniques for optimization, the system developed does not explore the use of deep learning to improve the recognition and classification of objects in real time, based on the use of Deep Learning in Vision Systems.
- The system developed by the authors is specific to a footwear production environment and focuses on optimizing specific tasks, but does not allow the system in question to adapt quickly to changes in the work environment.

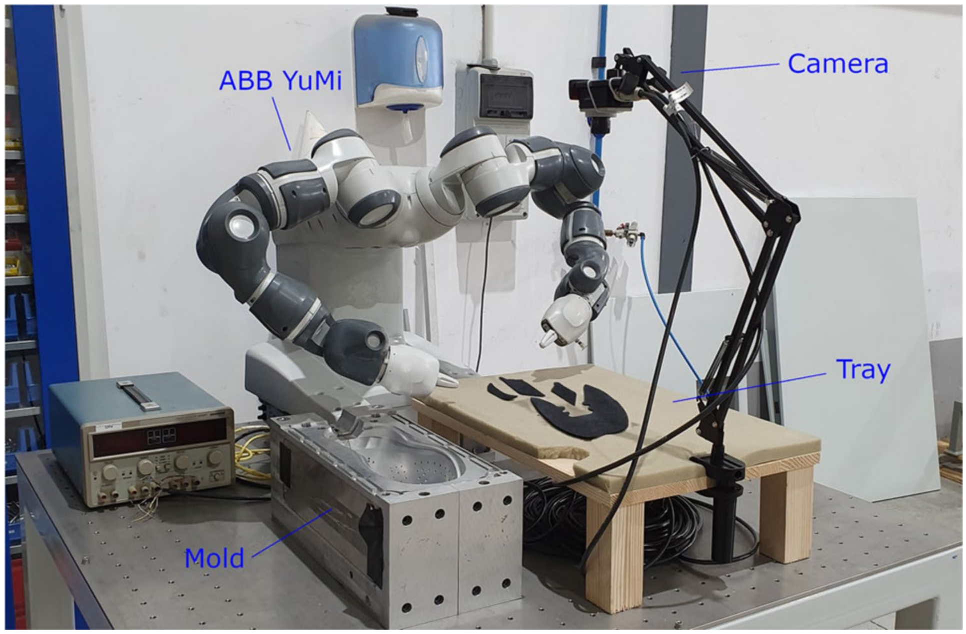


Fig. 3 Proposed system with an ABB YuMi cobot, camera, shoe mold and tray [14]

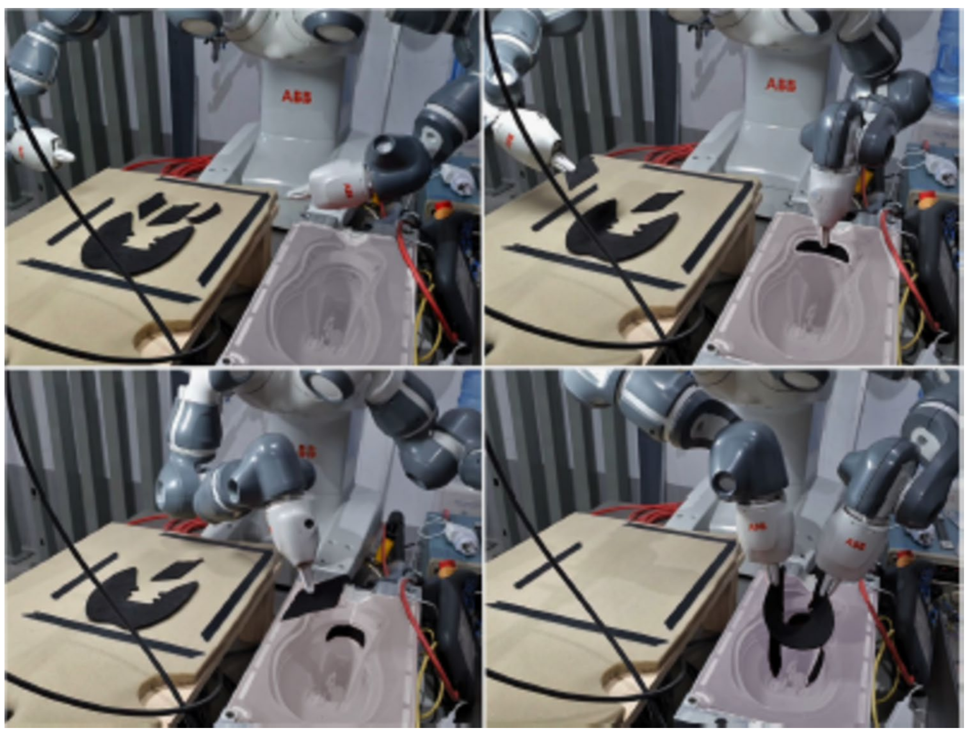


Fig. 4 Pick-and-place developed system in an experimental scenario [14]

- The article addresses the automation of complex tasks on a production line, it does not combine the use of AI and artificial vision.

2.3 Integration of AI in Collaborative Robotics within the Pick & Place Technique

Natanael Magno Gomes et. all, carried out a research study on the use of Reinforcement Learning (RL) to control collaborative robots (cobots) in pick-and-place tasks, which involve identifying, capturing and moving objects, represented in Fig. 5 [15].

The authors introduced the application of Deep Reinforcement Learning (DRL) to allow a cobot to adapt to changes in the position of objects in its work environment. The study they developed used a combination of a deep convolutional neural network (CNN) and a Q-learning algorithm to estimate capture positions (grasping positions) based on RGBD images (which combine color and depth data) [15].

The developed system was initially trained in a simulated environment, using different pre-trained CNN models, such as ResNext, DenseNet, MobileNet and MNASNet, to extract features from images. After training, the system was tested on a Universal Robots cobot with the UR3e model equipped with a two-finger gripper and an RGBD camera. Test results showed that the MobileNet-based model achieved the best performance, with an 89.9% capture success rate when manipulating a never-before-recognized object [15].

Based on our research, this work presents the following limitations [15]:

- Does not address the integration of a lighting source to improve image quality and, consequently, object recognition rates.
- The article does not discuss the handling of shadows caused by light sources, which is important to ensure uniform illumination and improve the reliability of object detection results.
- The authors addressed the adaptation of the robot to different positions of objects through reinforcement learning, but do not present details about considering the rotation of objects or about software adjustments related to the discrepancy between the predicted and actual position.
- The article does not extensively explore future advances, such as the development of adaptive algorithms or the expansion of recognition regions.
- Although the authors used deep learning to control the robot, its main focus was on task control and not on optimizing object recognition and classification in real time.
- The paper addresses the adaptation of robots to changes in the environment through reinforcement learning, but focuses only on controlled tasks. No adjustment was made to how the developed system could quickly adapt to dynamic changes in a factory environment with high production levels.
- The authors did not address how the combination of AI and vision can facilitate the automation of more complex tasks, such as quality inspection or the assembly of more fragile parts in an industrial environment.

2.4 Collaborative Robotics Systems (cobots) Equipped with Machine Learning-Based Vision

Xingyu Yang et. all propose a system with an automation framework represented in Fig. 6, for production in small and medium-sized companies (SMEs) using systems with cobots equipped with vision based on machine learning [16].

The research project focused on three challenges:

1—Increased visual perception of cobots;
2—Handling various tasks;
3—Rapid implementation of systems with cobots.

The developed framework included [16]:

- A Vision system based on Machine Learning implemented using the YOLOv5 algorithm and dataset [17] for object detection and a combination of Convolutional Neural Network (CNN) with Support Vector Machine (SVM) for product quality inspection;
- A Multifunctional and Reconfigurable Gripper capable of performing multiple tasks without the need to change tools, adapting to changes in the work environment;
- A Digital Twin in order to facilitate system development, debugging, and monitoring of the production process in question and in real time.

In this work there was a comparison between different convolutional neural networks (CNNs) models that were

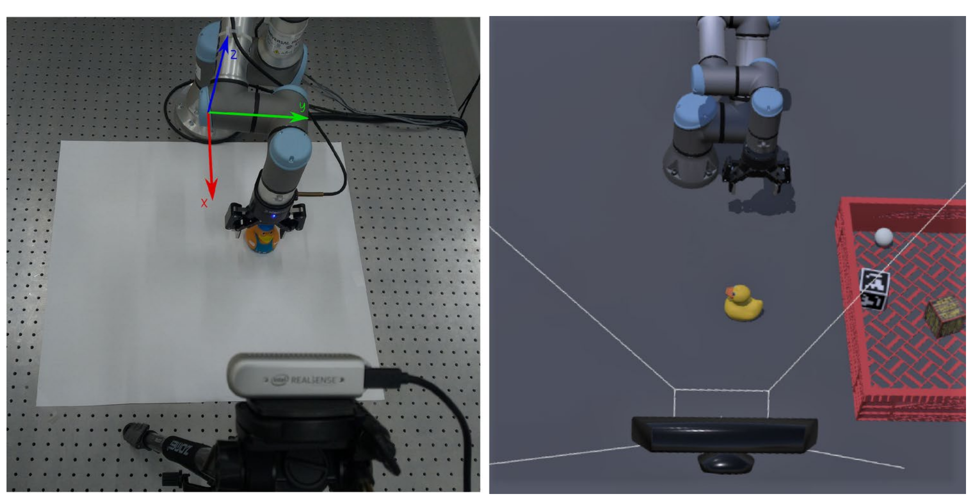


Fig. 5 Experimental & Simulation setup [15]

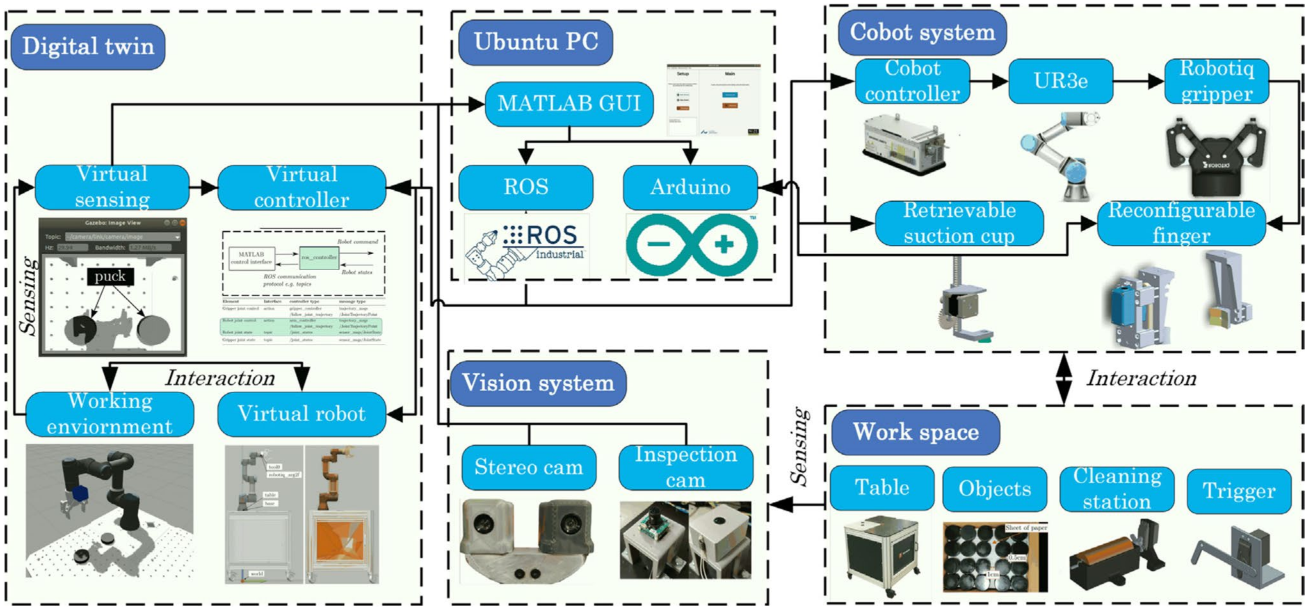


Fig. 6 The developed framework for ice hockey puck printing [16]

pre-trained and applied in a collaborative robotics system for pick-and-place tasks. These technologies discussed were [16]:

- DenseNet;
- ResNext;
- MobileNet;
- MNASNet.

In the research article in question, the main criteria used to compare these CNN models were [16]:

- Accuracy in recognition, measured by the success rate in identifying and picking up objects;
- Processing speed, being the time needed to process the images and decide the action to be taken;
- The computational requirements, which in this case were memory usage and training time.

Below in Table 1, has shown a comparative analysis of different image processing technologies and their suitability for object recognition in the context of pick & place applications.

Taking Table 1 into consideration, the following comment can be made [16]:

- Pre-trained CNN models, such as DenseNet and ResNext, showed average accuracy but with longer processing times due to their high memory requirements.
- MobileNet stood out with the highest accuracy (up to 89.9%) and the lowest processing time, making it the best choice for pick-and-place applications in collaborative robotics, where computational efficiency is important.
- MNASNet, although faster, showed significant variations in accuracy.

Based on our research, this work presents the following limitations [16]:

- Does not address the integration of a lighting source to improve image quality and, consequently, object recognition rates.
- The developed system needs improvements in the integration of lighting sources, as it still depends on simple approaches to solve the problem of shadow effects.

Table 1 Summary Comparison Table [16]

CNN model	Grasping Accuracy	Time processing	Computational Requirements
DenseNet	Medium (70–72.7%)	High (slower)	High (more memory)
ResNext	Medium–Low (64.7–67.7%)	High (slower)	High
MobileNet	High (76.3–89.9%)	Fast	Low (less memory)
MNASNet	Low (39.4–72.6%)	Fast	Low

- The authors focused their work on object detection, but limited the use to computer vision techniques without the integration of additional sensors.
- The developed system does not respond to the need for real-time adjustment in dynamic environments and has limitations in implementing AI for more complex tasks in which the environment is very dynamic.

2.5 Identified Gaps and Proposed Solutions

The following Table 2 presents the identified gaps in the proposed system and state-of-the-art systems.

The system proposed by the authors in this article introduces a series of innovations that respond to the limitations or gaps identified in Table 1 by the various systems addressed in the state-of-the-art.

By analyzing the main gaps in existing pick and place systems with cobots, we can identify the lack of solutions that allow:

- Design of a more flexible and adaptable system, especially in dynamic industrial environments, where precision and adaptability are critical;
- The inclusion of a specific lighting source, which resolves common shadow issues that affect object detection. This is a problem that is not adequately addressed by existing systems;
- Systems with the efficient integration of computer vision without being solely focused on optimizing trajectories;
- The integration of additional sensors and software to improve the robot's perception of its surroundings;
- Exclusive focus on task control with optimization of object recognition in real time;
- Systems that combine AI and computer vision for more complex tasks;
- Creation of adaptive algorithms for object recognition in real time.

Despite advances in research on integrating AI and image processing into cobot solutions, many existing solutions are still inaccessible to small and medium-sized businesses due to the high cost and complexity of implementation. The fact that existing systems do not allow effective control with the inclusion of a specific lighting source that solves common shadow problems that affect object detection is a gap with very effective importance, especially in quality control in Pick & Pick systems. place. The method proposed in this work aims to fill this gap by offering a cost-effective and scalable solution that can be applied in a wide range of industrial scenarios, from automated assembly lines to quality inspection processes.

The following chapter will present the hardware and software used, as well as the results obtained, demonstrating the effectiveness, applicability and innovation of the system proposed in this article.

3 Components—Hardware and Software

This section describes the proposals and solutions developed, and presenting justifications for all decisions made based on the initial challenges. The hardware and software solutions used in implementing the vision system will be presented, as well as all parameter definition methods used in image processing and control of the robotic arm to achieve the end goal.

Table 2 Comparative table of the proposed system and state-of-the-art systems

Identified Gaps in Existing Systems	Subchapter/System
Inadequate lighting for improving image quality and object recognition	2.1 (Muhammad Umar Anjum et al.)
Insufficient treatment of shadows caused by light sources	2.1 (Muhammad Umar Anjum et al.)
Dependence on controlled environments and difficulty adapting to complex or dynamic environments	2.1 (Muhammad Umar Anjum et al.), 2.3 (Natanael Magno Gomes et al.)
Lack of adaptive algorithms for real-time object recognition	2.2 (Jorge Borrell et al.), 2.4 (Xingyu Yang et al.)
Systems focused solely on trajectory optimization without efficient integration of computer vision	2.2 (Jorge Borrell et al.)
Systems designed for specific scenarios lacking flexibility for other industrial applications	2.2 (Jorge Borrell et al.), 2.4 (Xingyu Yang et al.)
Dependence on simulations without results in real industrial environments	2.1 (Muhammad Umar Anjum et al.), 2.3 (Natanael Magno Gomes et al.)
Limitations in integrating additional sensors to enhance robot perception	2.4 (Xingyu Yang et al.)
Exclusive focus on task control without optimization of real-time object recognition	2.3 (Natanael Magno Gomes et al.)
Systems that do not combine AI and computer vision for more complex tasks	2.2 (Jorge Borrell et al.), 2.4 (Xingyu Yang et al.)

3.1 Proposed System

The proposed system is shown in Fig. 7. It consists of a collaborative robot that will perform all pick and place operations. A vision camera is used to acquire images connected to a computer that will process them. The objects, rectangular blocks, to be manipulated are placed on a properly lit white background.

Then, the proposed system consists of the UR collaborative robot, responsible for pick and place actions, equipped with a tool capable of grasping rectangular blocks, a Zimmer collaborative gripper. An Intel D435i camera, from the D400 series, was attached to the end of the arm, ensuring visualization of the object to be identified. Integrating all systems, Neadvace's Niop software was adopted. The camera is used to acquire images, insert them into the system, which transmits them to the computer. Niop is responsible for all the logical processes, as well as the image processing and the procedures necessary to generate the results, transmitting them to the robot. The robot picks up the part and places it at a specific point.

3.2 Hardware

3.2.1 Collaborative Robot UR

Collaborative robots are increasingly present in industry. This growth is largely due to the possibility of being able to work in safe conditions alongside the operators [18]. Thus, depending on the security configurations considered, the cobots can operate in anywhere without being protected by a security fence, especially for this reason that they are easily integrated into a process, collaborating, side by side, with operators who perform less ergonomic tasks [18].

The collaborative robot used for the pick and place operation was the UR3e from Universal Robots. This cobot has a payload of 3 kg and a reach of 500 mm [19] sufficient to implement and test the application.

3.2.2 Zimmer Gripper

The collaborative robot itself only consist of an arm that will focus on the movement/trajectory of the cobot. Therefore, a suitable gripper is necessary to equip the cobot. The gripper used in the cobot is an HRC 03 2-jaw parallel gripper from Zimmer Group [20] (see Fig. 3 left). This gripper is compatible with a lot of brands and has integrated position sensing features that give the position of the jaws (analog) at any time, from completely open (0 V) to completely closed (10 V). The stroke for each jaw is 10 mm (repetition accuracy of ± 0.05 mm), the grasping force has a maximum of 190 N and a closing/opening time of 0.19 s.

3.2.3 Intel Camera

The D435i depth camera belongs to the D400 series, which is specifically designed to incorporate depth sensing functionalities in various applications. The justification for choosing this camera lies in its ability to capture depth, color, and inertial measurement system, which allows awareness of own movement.

The camera's depth feature presents opportunities to perform selection and positioning actions, making it a suitable tool for varied contexts. Additionally, the camera's ability to

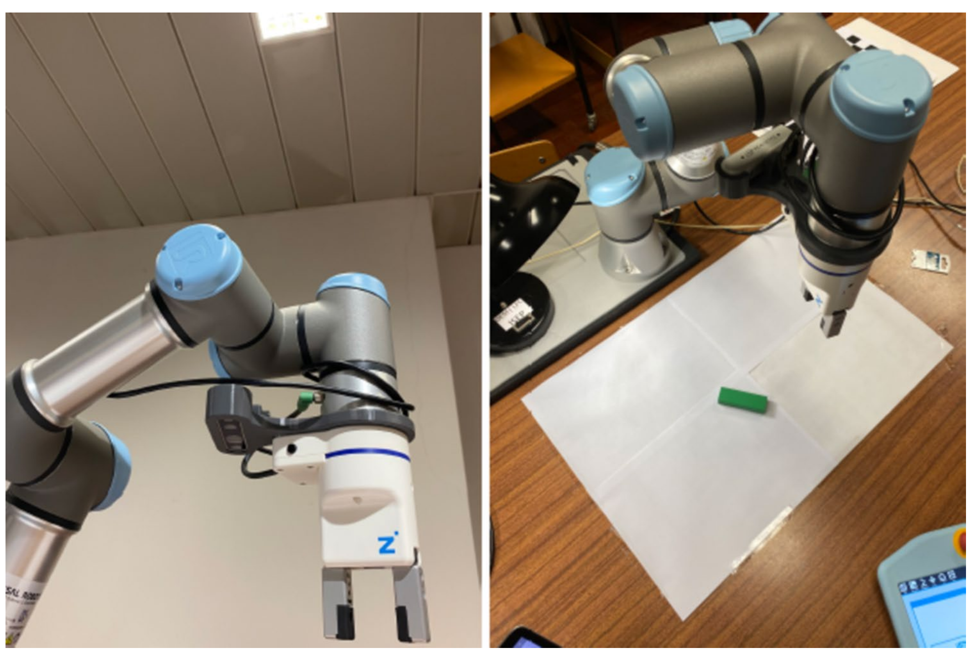


Fig. 7 Proposed artificial vision system

differentiate colors further increases its potential in complex systems and specialized applications. Camera's characteristics make it ideal for deployment indoors and outdoors, with an effective range between 0.3 m and 3 m. The depth technology used in this camera is the stereoscopic method, which uses two or more views of a scene to obtain depth information [21].

3.3 Neadvance Niop Software

Niop is a platform created by Neadvance [22]. Neadvance is a software development company dedicated to providing exceptional solutions for businesses of all sizes. Neadvance provides a wide range of software solutions such as automatic quality control, among others. Neadvance solutions are designed to improve efficiency, streamline operations, and increase productivity using computer vision and artificial intelligence software solutions. Niop software uses a drag and drop system that allows to create whatever easily and quickly [23].

The challenges to a Pick & Place job are:

- Perform pick and place of objects on a moving conveyor;
- Automatically adjust to the conveyor speed and unexpected velocity fluctuations;
- Perform the system calibration and communicate with the robot;
- Build the workflow in a structured way to be easily customized in the future.

In the system developed by authors, Niop software allows the coordination of pick-and-place operations, integration with sensors, and the execution of Pick & Place tasks with the Cobot.

The solution is based on 4 different components, showed in Fig. 8:

- The **Studio** is where the design of the program and the workflow is built.
- Once the program design is finished, it can be published to the machine PC where the **Engine** is running.
- On the **HMI** the end user can monitor the execution and adjust the parameters.
- Finally, the **HMI editor** is where the graphical user interface can be design to specific needs.

The interaction of all these components is illustrated on the following picture on Fig. 9:

- The studio communicates with all the components,
- it publishes the solution on the Engine and on the HMI.
- At the same time interacts with the HMI editor.
- The **HMI editor** can be accessed by the studio and all

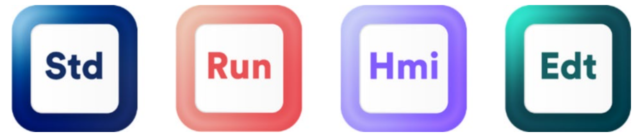


Fig. 8 Components from Niop Software [23]

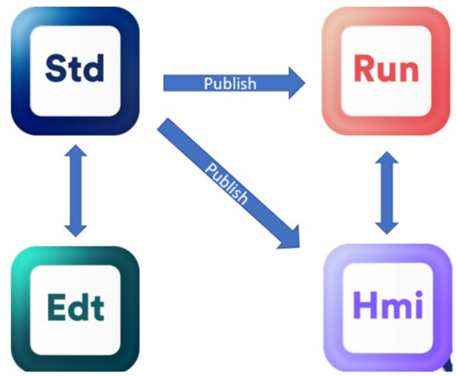


Fig. 9 Interaction of all the components in Niop Software [23]

- the modifications performed on the editor are
- automatically saved on the studio project
- The engine runs the cycle published by the Studio
- The HMI interacts with the engine by sending and
- receiving information from the end use

The functionality of each module used within Pick & Place tasks is exemplified below in Fig. 10, 11 and 12 [23].

1. **Calibration Module**

The calibration module identified in Fig. 6 is responsible for adjusting the system parameters to ensure that the robot's movements and image captures are accurate.

2. **Acquisition**

The acquisition module identified in Fig. 6 is responsible for starting the camera and capturing the images necessary for analysis. This manages the capture process, such as starting recording, obtaining the image buffer, and freeing resources after acquisition. These images are then processed to identify the objects to be manipulated in the Pick & Place task.

3. **Processing**

The processing module identified in Fig. 6 performs the analysis of the acquired images, applying the necessary calibration and processing the data to determine the position

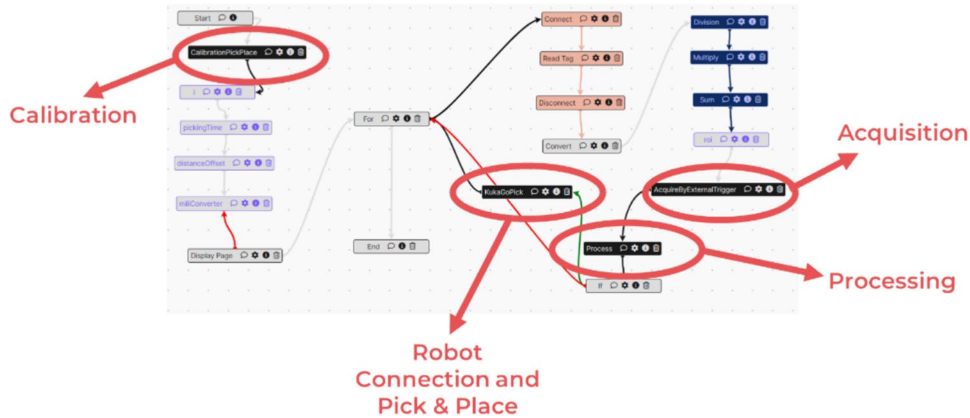


Fig. 10 Pick and Place−Main Workflow [23]

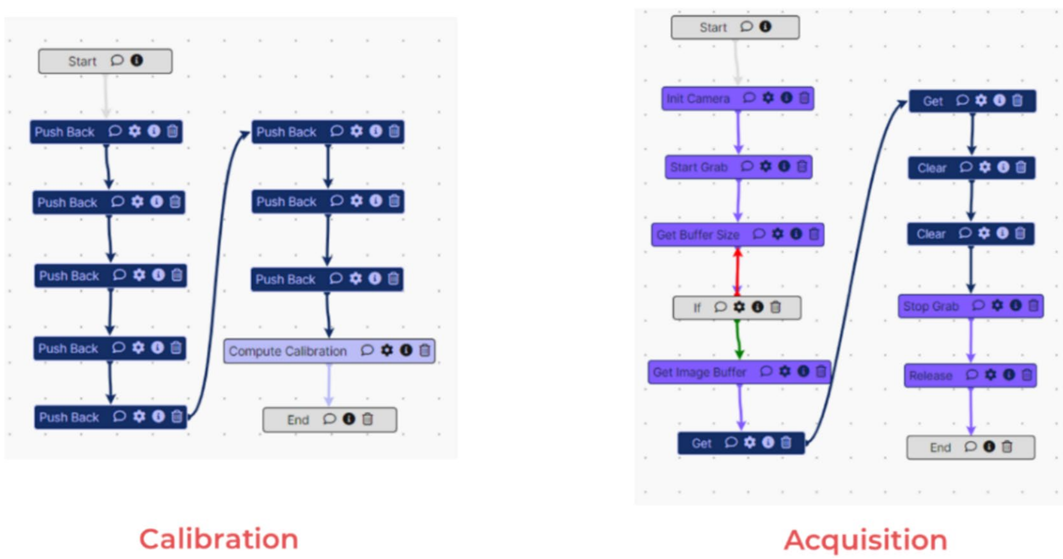


Fig. 11 Pick and Place−Sub Workflows 1 [23]

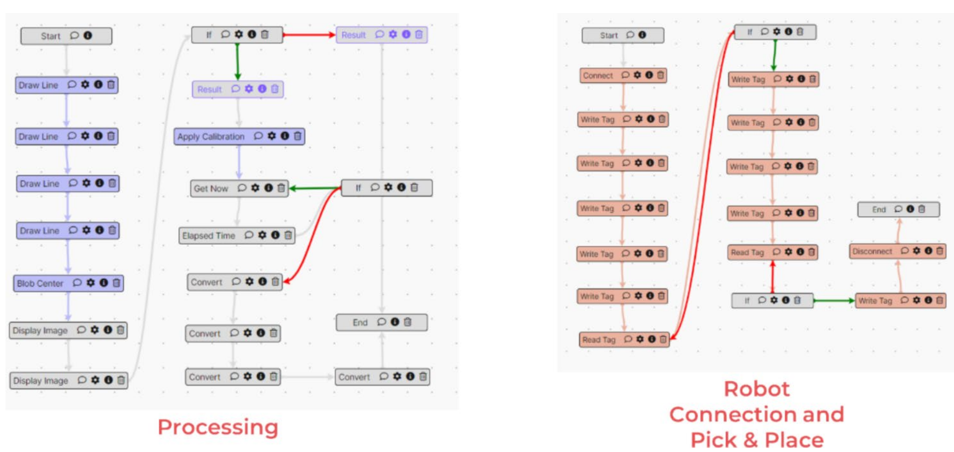


Fig. 12 Pick and Place−Sub Workflows 2 [23]

and orientation of the objects. This module is essential for converting visual information into commands that the robot can use to perform pick & place tasks.

4. **Connection with the Robot and Pick & Place**

The connection module with the Robot and for the Pick & Place task identified in Fig. 6, allows communication between the control system and the robot. This sends commands to the robot to perform the pick & place operation, such as picking up an object from a specific position and placing it in another location (Pick & Place operation). The module also monitors the execution of these tasks, ensuring that the process is executed accurately.

4 Materials and Methods

4.1 Computer Vision

Computer vision, initially used around 1970 [24], is a field of artificial intelligence that focuses on the ability of machines to recognize, interpret, and understand visual data from the world around them. This technology uses algorithms and automatic learning techniques capable of giving the machines the ability to interpret visual data, just like humans, so that they can make reactive and intelligent decisions based on the data. Computer vision systems can therefore detect and track objects, estimate distances, recognize faces, identify emotions, classify images, and even predict future events. These systems can work in real time and can be used in a range of applications, including robotics, surveillance, medicine, automotive industry, and consumer products [25].

A typical computer vision system is divided into the phases of image acquisition, image processing and image analysis [26]. These systems can be used to automate industrial processes, replacing humans [27], such as inspection, locating objects and many other repetitive tasks, so they are a fundamental in many applications. Computer vision systems are reliable systems capable of performing thousands of operations repetitively, far surpassing human capabilities [28].

4.2 Image Acquisition

Image acquisition is the initial stage in the process of identifying an object. At this stage, the electronic signals emitted by a sensor are transformed into a numerical representation based on the signals coming from a camera. So, the final quality of the image is the combined result not only of the lighting used, but also of the quality of the camera, i.e., the number of pixels. As such, since the number of pixels in the bed remains constant, lighting plays a fundamental role, as it directly affects the quality of the image. For this reason, lighting is of great importance in image extraction [29], particularly with regard to the reference of contours, shadows and even color accuracy. It should be as uniform as possible, and is considered a critical aspect in obtaining quality images [30].

Camera sensors are normally based on two main commercial types: the charge-couple device (CCD) and complementary metal-oxide semiconductor (CMOS), all based on the metal–oxide–semiconductor (MOS) [31]. The CCD works on the basis of a grid of sensors that take on a value proportional to the intensity of the light radiated onto the sensor pixels, with a high quantum efficiency [32]. CMOS sensors can capture the image via a rolling shutter (RS) or a global shutter (GS), with low power consumption, low supply voltage and high-speed image capture [33].

4.3 Image Processing

The perception that a human being has when looking at an image is very different from the perception that a computer vision system has. For a computer vision system, an image is a matrix of values in which each value represents a pixel, which in turn represents a color. The correct combination of these pixels creates an image that can be recognized as such by human beings. For humans to perceive an image, pixels must be combined with the three main colors: red (R), green (G), and blue (B), which make up the RGB value of a pixel. The color of the pixel is characterized by the values that each of the R, G and B components takes on in a range from 0 to 255. Thus, if the RGB values are all equal to 0 (0,0,0), the result is "black", while an RGB value of 255 (255,255,255) represents "white". So, these primary colors are represented by (255,0,0) red (R), (0,255,0) green (G) and (0,0,255) blue (B) [34], which combined produce a wide spectrum of colors.

Based on the above, it is clear that images need to be processed in terms of color pre-processing, greyscales, focus correction, contrast adjustment and noise reduction. Therefore, while pre-processing focuses on image quality and preparation, processing focuses on segmenting objects and extracting their characteristics, such as shape and color or area and perimeter. This is why the characteristics to be acquired will depend on the objects used and the final objective of the machine vision system [35]. Then, to do this, there are several algorithms that can be used and which have been developed to act on multiple characteristics, from color treatment options, noise reduction that interferes with the definition of the object's characteristics, as well as measurements to be taken and also identification of textures, among others [36–38].

Pre-processing usually consists of adjusting the image to greyscale, i.e., representing the light intensity information present in each pixel. Generally speaking, adjusting, or creating a greyscale image can be represented by the following expression:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

where 'Y' represents the grayscale value after the calculation. This conversion results from the need for a simplified representation of the color image. The conversion is due to the fact that there is only one variable in the grey scale, while in the color image there are three variables, as can be seen in Fig. 13.

Processing greyscale images is often much easier compared with color images, which is one of the reasons why it is often used. But a binary image is even simpler and easier to process than black and white images. Binary images contain pixels that can only have two possible values, usually black (0) or white (1). Therefore, the output value of the binarization will be defined by a threshold which is assigned a value between 1 and 0 if it is above or below, respectively. This methodology is very important for the object segmentation, as it allows the application of algorithms used in morphology and edge recognition operations. In general, the binarization of the image $I(i,j)$, using a greyscale image, and considering a threshold value (T), can be represented by:

$$B(i,j) = \begin{cases} 1 \ if \ I(i,j) \geq T \\ 0 \ otherwise \end{cases} \quad (2)$$

Therefore, image binarization will be achieved by considering a user-defined global threshold, so that any pixel value above the threshold will be converted to white (or 1 in binary), while any value below the threshold will be converted to black (or 0 in binary). For instance, if the threshold value is set to 100 and a pixel in a grayscale image has a value of 120, it will be converted to white (1) in a binary image, see Fig. 1 (right). However, in certain cases, using a global threshold may not be the best choice. This uncertainty will manifest itself in a lack of uniform illumination, which can lead to part of the object being categorized below the global threshold and another part above, making segmentation difficult. To overcome these limitations, thresholding with local values is used to improve the segmentation of images of objects with low illumination [39].

Another important phase of image processing is the need to make adjustments to the image to improve it for later use or to make it sharper. These adjustment operations are generally referred as morphological operations and are usually used to eliminate some irrelevant forms and keep the essential. Of all the morphological operations, the most widely used and simplest are the "erosion" and "dilation". They have the same operating principle, but work in an antagonistic way, expanding or reducing the existing shapes in the images. "Erosion" consists of combining sets by reducing the shapes through vector subtraction to the values of the elements of the set, that is, comparing the value of a black pixel with all its surrounding pixels and, if they are all black, it will remain black, otherwise the pixel will be white. On the other hand, "dilation" will work to enlarge the shapes, being an operation contrary to erosion, as a sum is performed between the elements of the set, that is, it will make all the surrounding pixels black if the pixel is black [40, 41].

Figure 2 shows what morphological operations can do. The top images (a, b, c) show irregular outlines and scattered small white dots. These small white dots do not add any extra value to the image. With the morphological operation it is possible to clean these points and make the sides of the rectangle flatter, i.e., morphological operations were useful for increase the accuracy of detecting edges in the image. Finally, it is also shown how the combination of the previously mentioned techniques affects the final image quality depending on the threshold value and morphological operations [42, 43]. The threshold value increases from left to right, from 50 to 120, see Fig. 14 lowest row.

5 Proposed Solution

This Section describes the solution for giving the robot the ability to pick up rectangular blocks and place them in a predetermined position and location, as well as the justification of the decisions made. The Section is divided into four parts, including object detection, calibration, data communication, and selection and placement operation.

In this system, was use NIOP software for object recognition system, which is a closed platform using Artificial

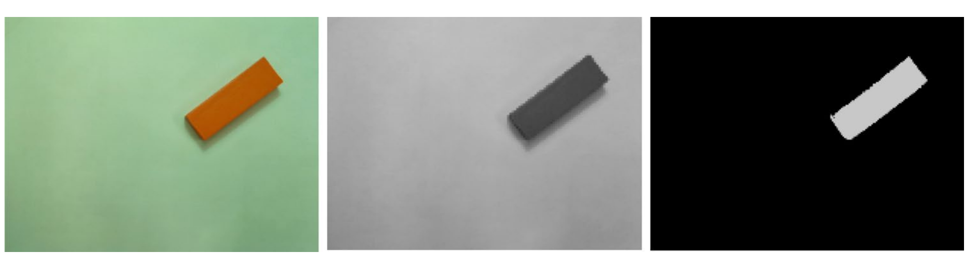


Fig. 13 Color image (left), grayscale image (middle) and binary image (right)

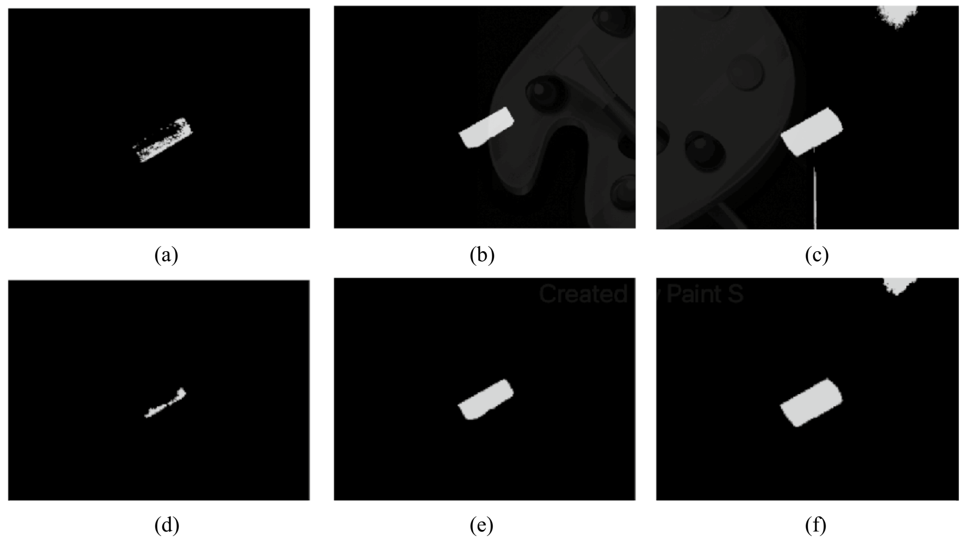


Fig. 14 Binary images with threshold value 50 (**a**, **d**); 75 (**b**, **e**) and 120 (**c**, **f**). With on the lowest row (**d**, **e**, and **f**) the binary images with morphological operation

Intelligence (AI) algorithms. NIOP offers a number of advanced features, including image processing, segmentation and object detection modules. The specific AI-based algorithms used by the system are not available to the user and cannot be changed as the software is low-code.

The use of AI at NIOP has allowed:

1—Detect patterns or anomalies in used parts;
2—Classify objects or products based on visual characteristics (such as shape, color and size).
3—Recognize complex patterns, such as marks in images that are difficult to detect only with traditional image processing algorithms.
4—It allowed decisions to be made automatically based on the processed images, optimizing processes such as the pick and place task and the inspection of colored parts.

Integration with the robotic system was carried out via the TCP/IP communication protocol via Workflow, which allowed sensing data such as position and orientation coordinates to be used directly to control the robot's pick and place tasks in real time.

The solution is supported by the software Niop that uses workflows to organize and isolate parts of the project, making applications clearer and organized. Figure 4 represents the 'Main' workflow of the project. In this workflow everything is connected. Each workflow will be called from this main workflow, and the 'Start' and 'End' are respectively the beginning and the end of the application. Therefore, the main objective of the whole project is to make the center point tool (TCP) of the robot coincide with the 3D center point of the object and according to the correct orientation. To do this, it is necessary to obtain the location of the object using Niop.

The application in Fig. 15 starts with "Data Collection" to extract all the useful information from the image that is needed for the pick and place operation, evolving into the "DataSending" workflow. DataSending workflow sends the positioning data to the robot using a Modbus communication protocol. The application sets a bit high, using the "Write Single Register" tool, to inform the robot that it can start the pick and place operation. Then, gives the robot some time to react, waits a while, before setting the same bit low again. Finally, the software disconnects from the robot and ends the application.

5.1 Object Detection

The object detection process uses a systematic approach involving the use of techniques and tools to obtain the relevant parameters. The most important parameters required for a correct choice of robot are the coordinates of the center point and the rotation of the object. To this end, various methodologies were applied to ensure that the parameters obtained are reliable and can be used to draw accurate conclusions. "DataCollecting" workflow is responsible for capturing images and presenting the information collected in a human–machine interface (HMI). The workflow captures the image to obtain position and rotation information.

5.1.1 Image Acquisition

As can be seen in Fig. 15, building the application involves using tools, which, when developed and combined, create an application adapted to the user's needs. The software evolves according to the direction defined by the blue and green

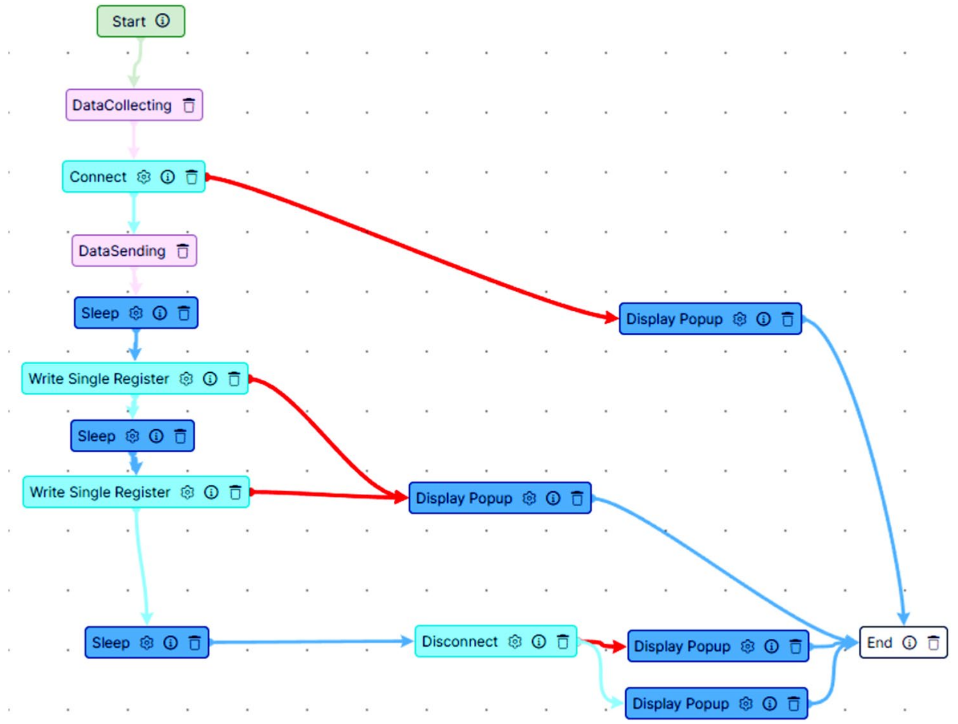


Fig. 15 Main workflow

arrows. The red arrows indicate the path that the software should take when an error occurs.

The Intel real sense D435i depth camera, perfectly identifiable by Niop in a broadcast action, was used as the image capture element. The information collected by the application is extremely important because it is on the basis of the camera's serial number to identify the width and height of the frames and the frames per second. The default values for width, height and frames per second used for the D435i camera are 1280, 720 and 30, respectively. Therefore, once all the image and camera parameters have been set, a color and depth image are captured, displayed on the HMI, and stored as a global variable so that it can be called up later in each workflow.

With the workflow called "RotationAndLocation" assigned to the "DataCollecting" workflow, it is possible to obtain the 2D center point and the first part of the object's rotation, Fig. 16. This figure illustrates two areas of code responsible for recognizing all rectangular shapes: area 1, and the dedicated area 2, responsible for recognizing the largest rectangle in the list of recognized rectangular shapes.

In area 1, we start by processing a copy of the color image taken earlier, keeping the initial image intact. Next, and as described above, the color image is converted to grayscale, "Convert Color", followed by its visualization. In the meantime, and at this stage of processing, it is important to define the threshold value that will be used in the binarization of the image. This value, set in the "Absolut" tool, will be initially assume a threshold of 70, and the image will be processed in the "Threshold" tool. The result can also be visualized on the HMI. Once the image has been binarized, morphological processing tools are applied to clean it up. The "Morpholoy Ex" tool cleans up the image, preparing it for the next stage of defining the object's contours, the "Find Contours" tool, storing them in a list of points.

Now the workflow is transferred to the processing area 2 via the "ForEach" tool, which enters the "point list" defined earlier. The "Min Area Rectangle" receives the list of points and checks whether it can find a (rotated) rectangle shape in it. If checked, it defines the smallest rectangle that fits the contour provided by the "point list". This check also involves validating a (rotated) rectangle by calculating, with the help of the "Point Box" tool, the perimeter of the rectangle's outline. The contour perimeter will on its turn be used by the “Contour Area” tool to calculate the area of the rectangle. Now the rectangle for initial “list of points” is defined as the outline of the rectangle in the form of a list of 2D points and as the largest of the rectangles.

The 'If' tool is used to find the biggest rectangle. When found, it is compared with the area of the biggest rectangle (which is zero at the moment), automatically loading it and setting it as the biggest by replacing the current one, which is zero. The comparison continues until all the rectangles found have been compared with the biggest rectangle stored. On the other hand, if the area of the largest rectangle is not between 3000 and 7000 pixels, the software will first add 2 to the threshold value and then count an integer

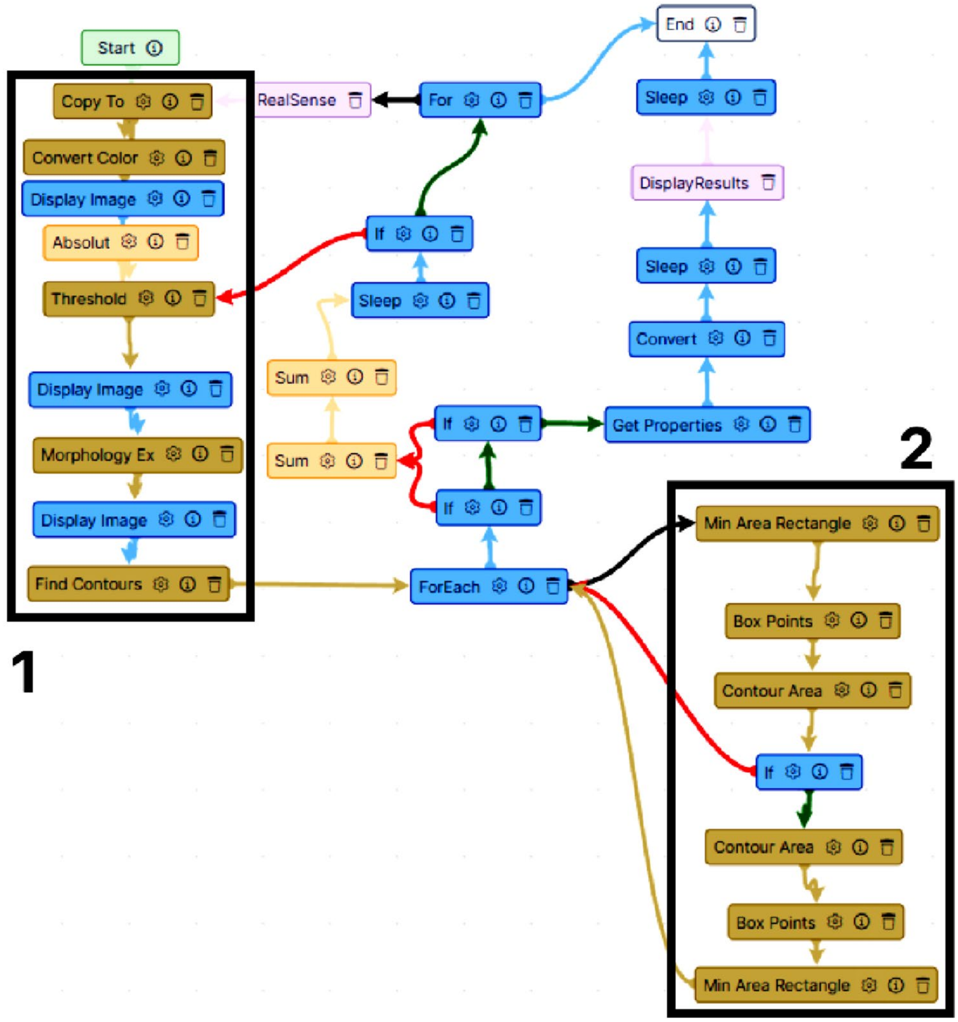


Fig. 16 "RotationAndLocation" workflow

safely up to 1. However, other variables can affect the image, including lighting, which can sometimes change, so we set the threshold value not as a constant, but as a variable. This simple change made the software much more successful. In addition, the safety integer number introduced was used as a brake to prevent the threshold value from ending up in an endless cycle. If this safety integer is not exceeded, the software repeats the steps described above from the "Threshold" tool. If the safety integer is exceeded, the software activates the "For" tool and repeats the entire process.

If this is true, the software goes to the 'Get Properties' tool (see Fig. 16) and gives the 2D center point and rotation angle of the largest rectangle that was found. If the object's center point has already been found, you can't say the same about the rotation since it hasn't been properly defined yet. So, the resulting angle is stored as a double, but for further use it is more logical to store this value as an integer because when sending this to the robot it needs to be an integer.

5.1.2 End-effector Rotation

Rotational information is essential in picking up objects accurately. This involves analyzing the orientation of the object in comparison to a standard orientation, because the two handles of the robot's hand need to fit over the small side of the rectangle to ensure picked up. So, for this work, the position in which the angle equals zero (or 180°) is defined as the object being placed vertically with the shorter side aligned with the x-axis and the long side aligned with the y-axis. And if the object rotates clockwise around its center point, the angle will increase until it reaches the vertical orientation again. Then, to find the correct rotation, it is important to know how the software calculates the system's coordinate. To do this, we used the coordinate system shown in Fig. 17(a).

However, when using this coordinate system, it will be impossible to obtain the correct rotation of the arm since, for example, if the result were an angle of 45°, we would always have 2 possibilities for an angle, and that results in incorrect data. Then, in order to get a unique angle for every rotation of the object, the shape of the object is also very important so, due to the object's rectangular shape, it has two lines of symmetry. The one line of symmetry shown in the Fig. 17(b) is the one who is most significant. It means that the rectangle will remain in the same location if it is rotated 180 degrees. Because 180° is essentially the same as 0°, this has the advantage that only the rotation from 0° to 180° needs to be defined, represented in Fig. 18.

Then, two additional points had been assigned next to the object's center point to help identify the direction of rotation of the object. Using the list of 2D points that represent the borders of the rectangle, each one is divided into its x and y components, after which the software checks whether the y coordinate obtained is greater than the previous one. If this comparison is verified, the software saves the biggest y-point together with its corresponding x-coordinate, Fig. 19(a).

If not, the software will try the find the smallest value for x. The application checks if the value of the x coordinate does not equal zero, because if the points are incorrect and equals zero it may not be able replace the smallest x value. However, it is almost certain that the x coordinate of the smallest x value does not equals zero except when it is exactly located at the left border of the image. So, the application takes the second smallest x value, which will

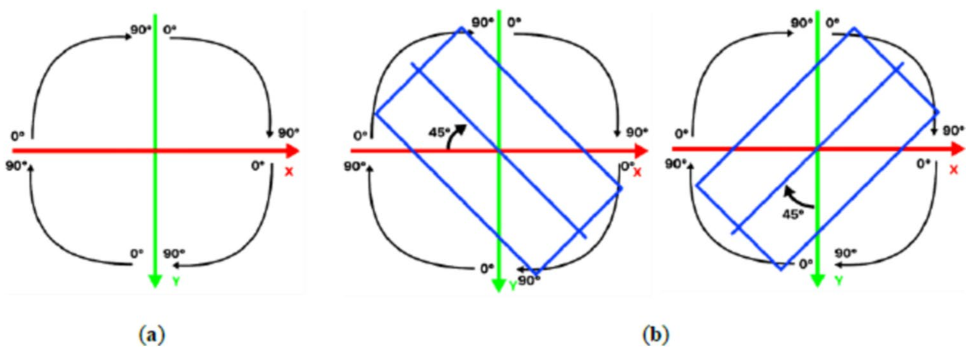


Fig. 17 (**a**) Original coordinate system; (**b**) 45° angle problem

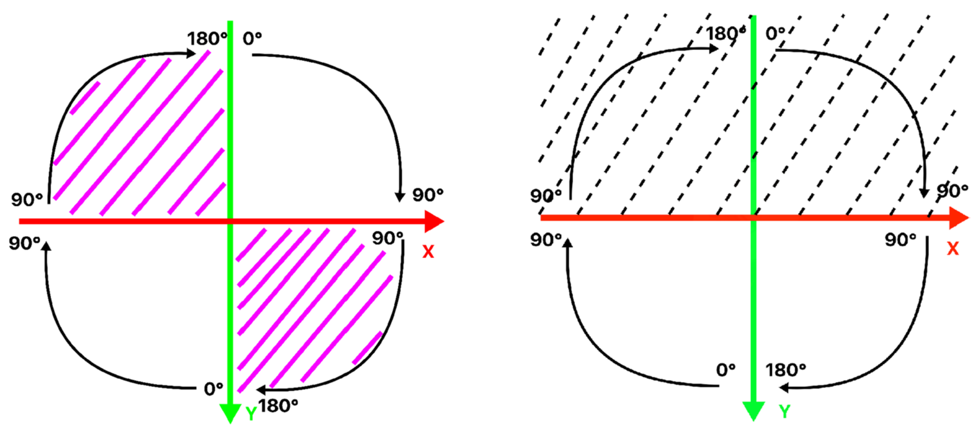


Fig. 18 New coordinate system

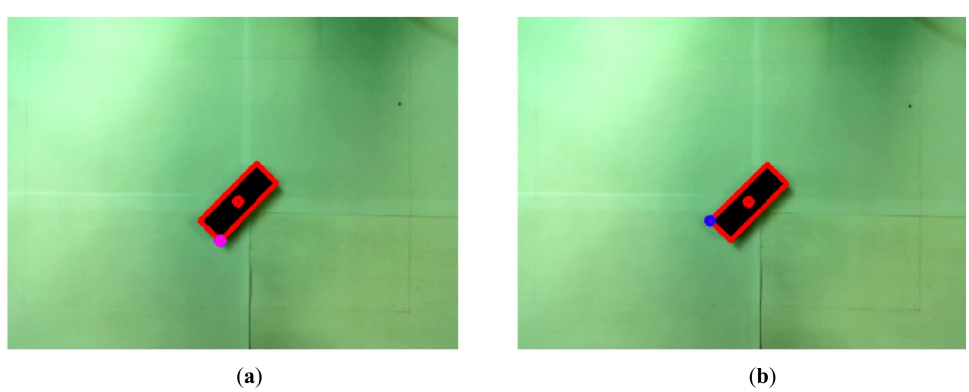


Fig. 19 (**a**) Point with the biggest y value; (**b**) Point with the smallest x value

be located very close to the value of zero. In that case, software compares the value with the value for the smallest x. If the value is smaller than the smallest x value found so far, is replaced. At the start of the application the smallest x receives a value of 100,000 since every first x value could replace this value. After every point has been checked, the software will find the 2D point with the smallest x value, Fig. 19(b). The points that we calculated, are the corners of the rectangle.

Once you have this information, you can use the x and y coordinates of the center point, the most left corner, and the lowest corner as information to find the exact angle of rotation of the object. The positioning of these points is calculated taking into account the distance between the y coordinate of the bottom corner and the y coordinate of the center point, i.e., the vertical distance (Fig. 20(a)). The distance between the y coordinate of the bottom corner is now subtracted from the y coordinate of the most left corner to obtain a result known as the 90° correction, Fig. 20(b).

With these considerations as a starting point, the application carries out various positioning checks, asking whether or not it is necessary to add 90° to the calculated value. These calculations make it possible to check that the rectangle is in a vertical position, i.e. with an angle equal to 0°. To do this, it is necessary to check that the "vertical distance" is greater than 55 pixels (Fig. 20(a)) and that the "90-degrees correction" is less than 0.60 pixels (Fig. 20(b)). When both conditions are validated, the angle obtained is set to zero, otherwise the application continues its check.

Other checks are made to determine the orientation of the object. If, for example, the angle is within a range of less than 20°, and the "vertical distance" comparison is less than 50 pixels, 90° is added to the angle obtained. Otherwise, if the angle is greater than 20°, the application checks whether the y-coordinate of the most left corner is smaller than the y coordinate of the center point this makes the necessary adjustments. The checks continue by making further comparisons of the vertical distance, for example, if it exceeds 38 pixels, 90° will be added to the angle obtained, and so on. Figure 21 shows various rotation possibilities.

5.2 Calibration

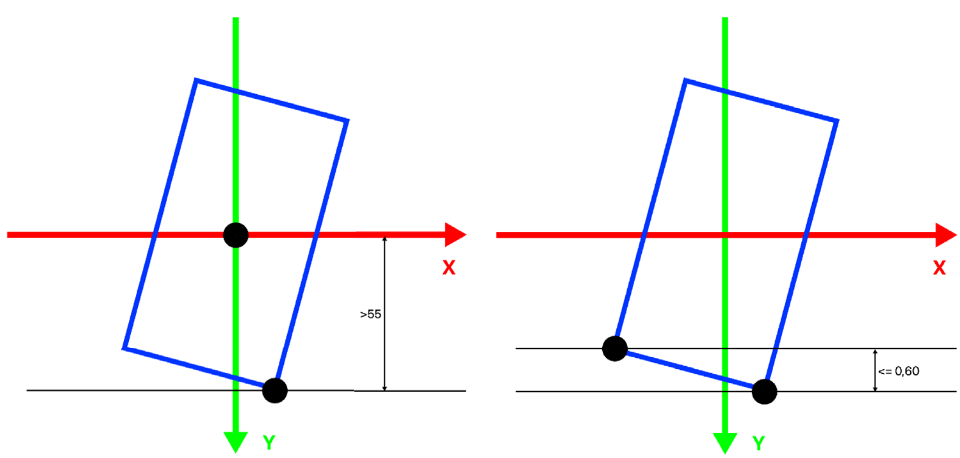
Calibration is fundamental to the correct functioning of the pick and place action, and it is impossible to carry out the intended operation without it. So, it is necessary, in this case, to translate the coordinates of the center point of the objects, obtained by the camera, into the coordinates of the robot (in mm), i.e., to establish the relationship between the coordinates shown in Fig. 22.

5.2.1 Calibration Method

The pick and place system were also calibrated using the Niop application. To do this, the camera was placed on the work area pointing to the six calibration dots placed on a white paper (Fig. 22(c)). The first goal of this process was to collect the coordinates, for the software, of all the dots and then by the robot by manually placing the TCP of the robots on the dots. The software used the "RealSense" workflow to capture the images, based on the concepts presented later, Fig. 12.

The "GetCircleCoodinates" workflow (Fig. 23), following the steps defined above, transforms the color image into a binary image with a fixed threshold of 70. Following the previously defined procedures, the list of contours is created, and the circularity of the dots has been confirmed, resulting in the creation of a 2D center point which will then be divided into its x and y components. As a result of this processing, Fig. 24 shows the binary image resulting from the transformations and de coordinates of de points. The table below the images, each column represents the coordinates

Fig. 20 (**a**) Vertical distance; (**b**) 90° correction



(a) (b)

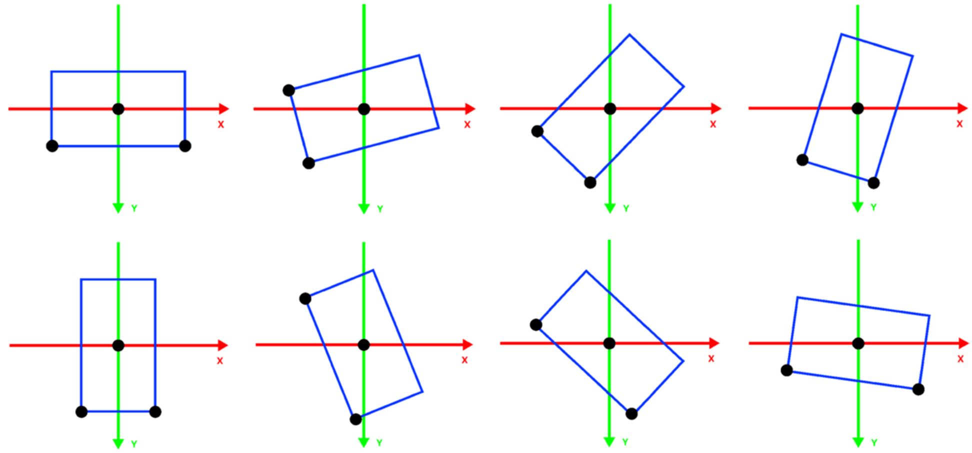


Fig. 21 Examples of rotation

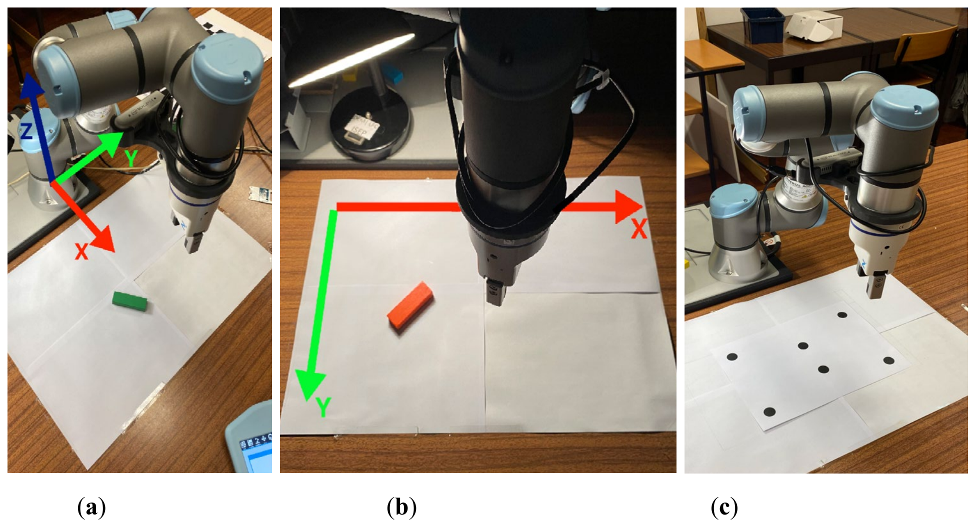


Fig. 22 Coordinate system: (**a**) of the robot; (**b**) of the camera; (**c**) placing calibration

of a point, while the first line shows the x coordinates and the second the y coordinates.

5.3 Data Communication

The Niop application was created as a development platform providing dedicated tools, such as the communications module, which can be easily used to develop Modbus communication. Its configuration requires knowledge of the starting addresses, the number of communication ports or registry values. Register values are integers, such as the 2D center point, angle value or even decimal numbers. The starting addresses and port numbers are selected from the universal robot's Modbus register map [22] indicating where input and output data can be written and read. This interconnection between Niop and the robot, a simple and transparent switch was used to ensure the computer-robot network isolated from the Internet. The connection between these is made by Modbus TCP/IP (Transmission Control Protocol/Internet Protocol) over Ethernet. Connection between the robot and the computer is made through the robot's IP (192.168.0.5) and the Modbus TCP communication assigned to port 502.

5.4 Pick and Place Robot Operation

Universal robots use their own graphical programming interface named PolyScope. PolyScope will provide build in functions and URCaps that allow to easily build applications. URCaps is a platform for UR accessories like for example the Zimmer group gripper that is used in this project.

The robot's pick and place program are responsible for managing the opening and closing of the gripper as well as its initial positioning. The picking action starts as soon as a bit called "digital_out[0]" is set high. At this point, the robot controller receives the positioning and orientation data of the object displayed through the ports shown in Table 3. This data defines the coordinates of the x and y components of the object's center point.

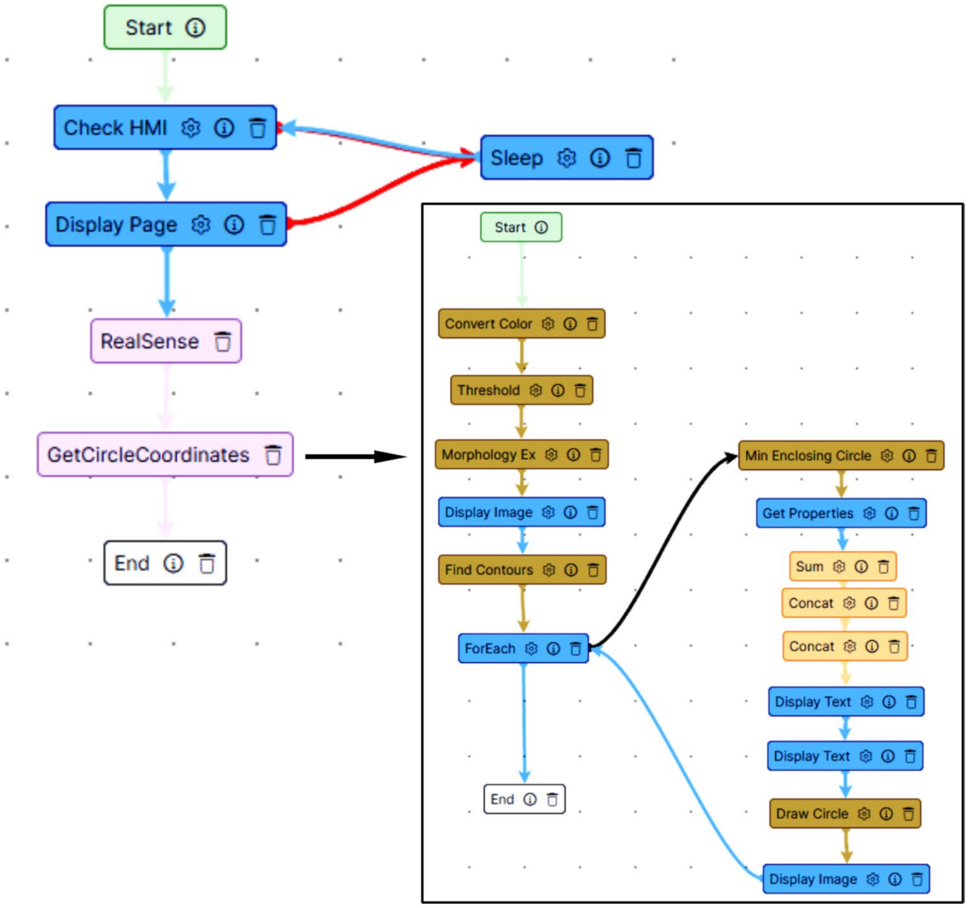
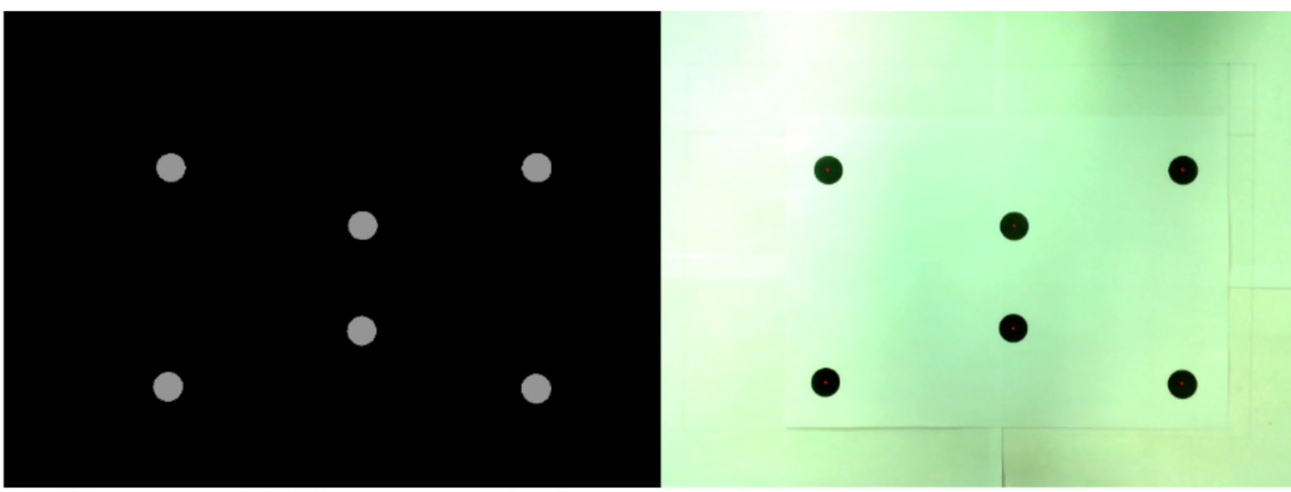


Fig. 23 "Main "and "GetCircleCoodinates" workflow calibration



518.629028	163.227524	350.148163	351.107147	519.260010	165.926392
375.983856	374.061798	320.000000	218.500000	162.539993	162.560913

Fig. 24 Example of the calibration

Table 3 Robot read register

Coordinate	Port_register
X	read_port_register(128)
Y	read_port_register(129)
angle	read_port_register(130)

Table 4 Object center position

Variables	Values
X_cp	0.001*X
Y_cp	0.001*(Y-1000)
Z_cp	0
RX_cp	3.14
RY_cp	0
RZ_cp	0

However, immediately before picking the object, the robot program must convert the data received from Niop (in mm) to its working units (m) by transferring it to the variables shown in Table 4. The rotation of the TCP and the value of the z coordinate only receive initial positioning information.

To continue, the robot must recognize the current positions of the joints, using "get_target_joint_positions()" functions, and store them in a vector $p[\ldots]$ with its six variables $p[x, y, z, Rx, Ry, Rz]$. The convergence of the software's angle with the robot's z axis rotation was ensured with the "Angle_offset" parameter, which was set to 24.25°. Since the robot is stationary above the object, this convergence ensures that the z axis rotation is in the correct position to pick up the object, so all that remains is to lower its z coordinate. The robot continues its movement towards the object's delivery point.

5.5 Algorithm for Pick and Place Operation

For the final part of this system, the robot needs to be programmed so it knows what do with the data it receives. The Fig. 25 shows the overview of our algorithm on the way the robot functions work together with the Niop software. The robot will perform a permanent loop and the software will need to be started manually each time. Together they will shape the project.

6 Discussion and Results

6.1 Light Effect

There is no guarantee of obtaining clear values when processing images based on the threshold value alone, as the binary image may not be of sufficient quality for shape recognition. The addition of a light source has significantly improved image interpretation, making it more accurate and with recognition rates close to 100%. Lighting reduces shadows and makes colors more vivid, making a difference in image interpretation. On the other hand, we must always bear in mind that the threshold value component in the production of a meaningful binary image is preponderant in the threshold-illumination combination. Thus, a pixel in a grayscale image must be white (0) in the binary image if it is greater than the threshold value. And when we consider the higher results for the lower threshold values for the darker image, this makes sense. Only the darkest pixels become white when the threshold value is too low. The image without a light source is the darkest. Therefore, the image without a light source will perform best with the lowest threshold values. This also applies to the image with the light source and high threshold values.

Table 5 in chapter 6 and sub-chapter 6.2.1 results, makes this very clear. How higher we go in threshold, by looking at the image without a light source, the more the algorithm also will turn the less dark pixels into white (0). This makes the binary image useless for further utilization. The most important part of this table are the results and for the results it seems that the best results are coming from the binary image created by the image with a light source. After discussing this table, it is fair to conclude that in general the better outcomes are coming from the binary images created by the image with the light source.

During system development, uneven lighting and shadows cast by objects presented significant challenges to object recognition. These challenges were addressed with the addition of a controlled light source, which improved the distinction between objects and the background, resulting in greater accuracy in capturing the binary images required for subsequent system processing.

6.1.1 Shadows

The addition of a light source is beneficial for object detection results. However, it can cause shadows which can lead to misleading results. These shadows are created by the angle at which the light falls on the object compared to the camera's point of view. Therefore, in order for shadows to be eliminated or reduced to almost zero, it is necessary to create a distributed, uniform, and enveloping illumination of the object. Figure 26 shows the effect of light position.

6.1.2 Results

Location and rotation are the important factors that are required to carry out a successful grab. It is reasonable to point out that while the location is not always accurate, the

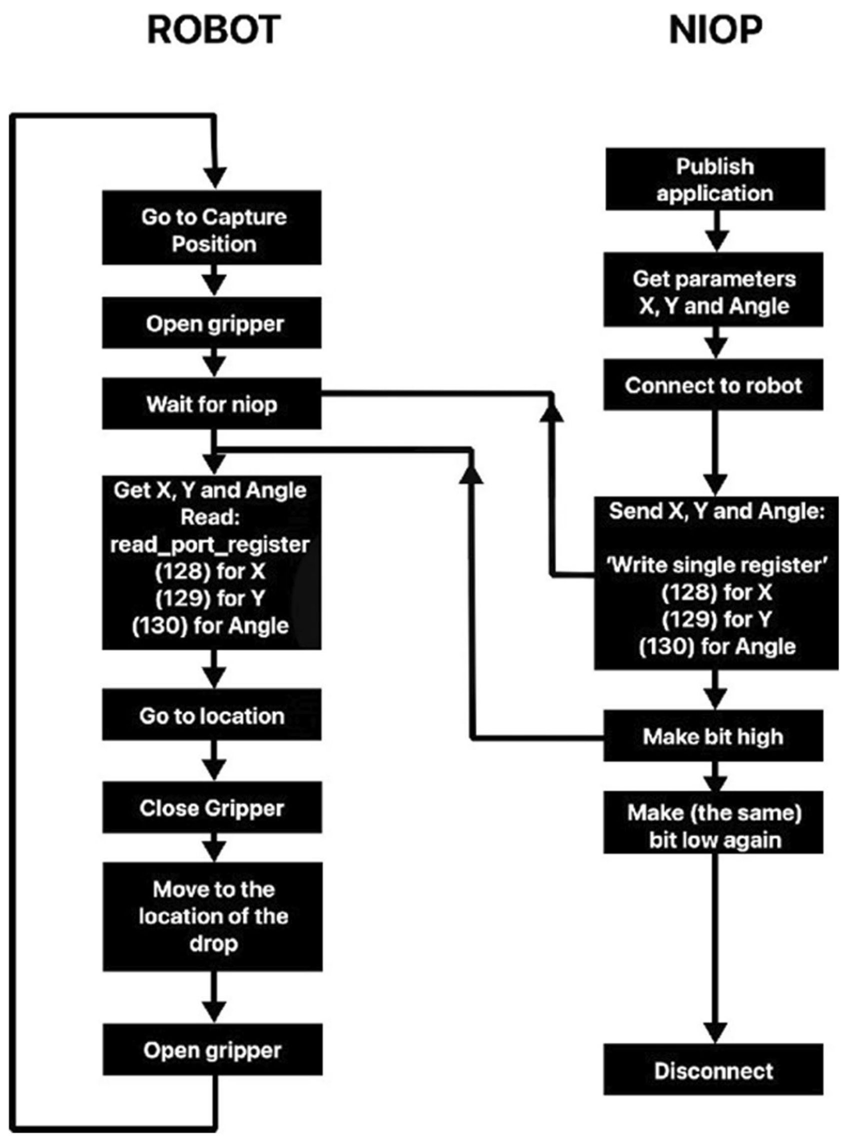


Fig. 25 Algorithm for Pick and place operation

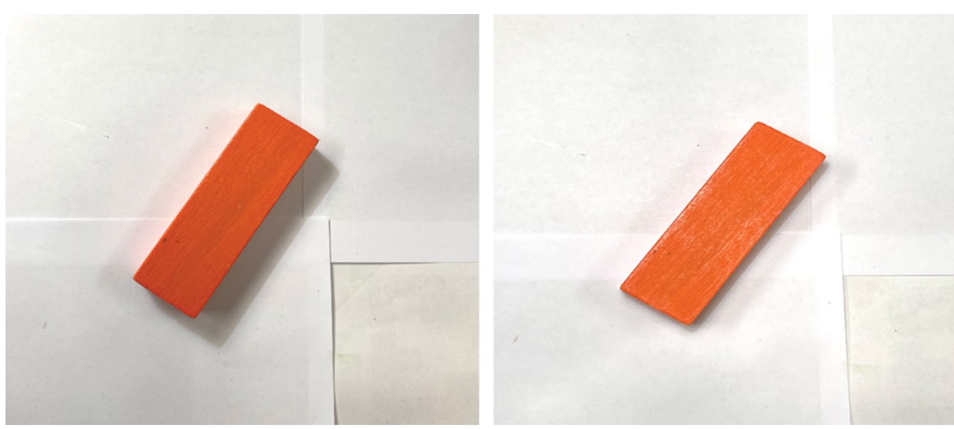
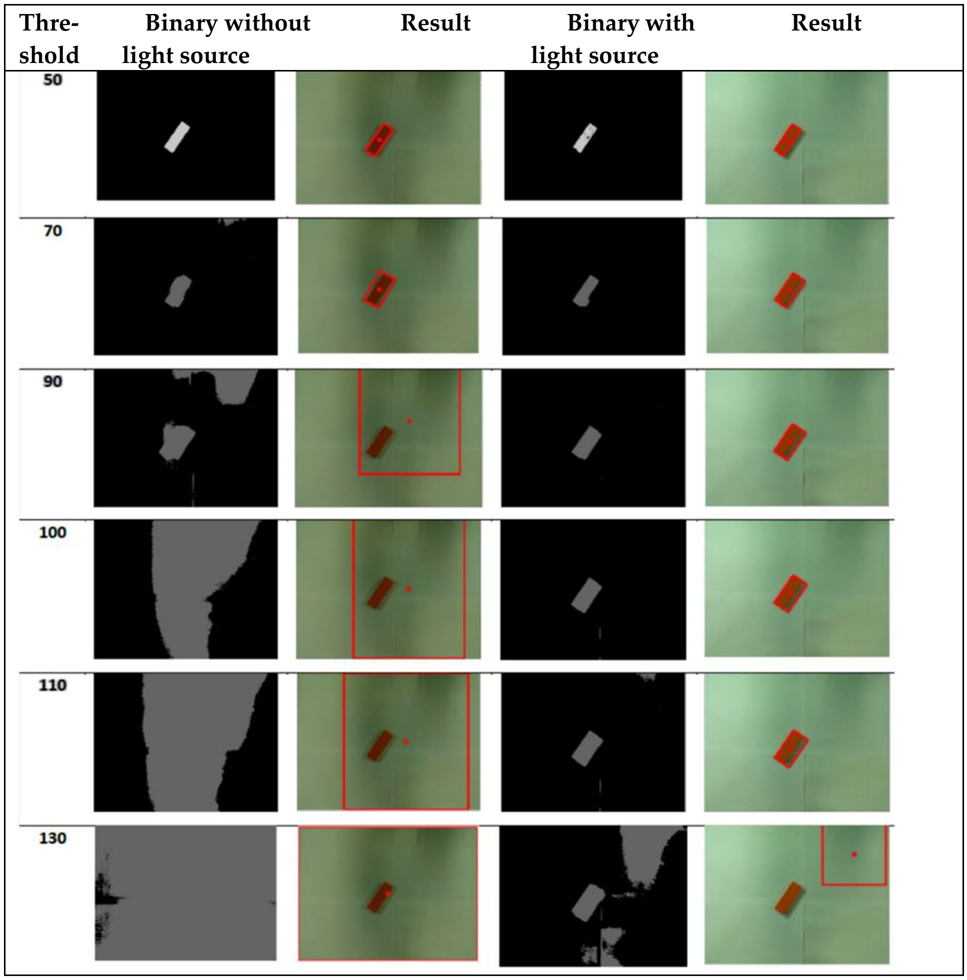


Fig. 26 Light placed in the left corner and next to the camera

rotation is almost always close to flawless. Even though the deviation of the location is always only a few millimeters, it prevents the pick and place operation. This can be caused by some already discussed but also some other elements. Even good results in the software can cause a deviation on the location of the robot. The main fails are caused by the side

Table 5. Results of a binary image with a changing threshold for an image with and without a light source



of the object, the shadow of the object and sometimes even with the use of correct results given by the software. These results are presented in Table 5 below.

Analysis of the results showed that the addition of an adequate lighting system is important in improving the system's accuracy. Without this improvement, shadows caused significant errors in object identification and rotation calculation, which limited overall performance. Although the adopted solution has increased accuracy, some limitations remain, such as the system's sensitivity to sudden changes in light intensity, which can be explored in future work.

6.2 Position of the Object in the Capture Region

The camera has a certain view which matches a certain picking region. Throughout the tests of the program, it came to the attention that how further away from the middle of this region how bigger the chance was to get an incorrect result. The main area where the pick and place operation cannot be carried out is in the higher region since this is where the robot is physically unable to reach the object without colliding with itself.

Another potential problem is when the object is more to the edge of the region, sometimes the side of the object is visible. This can cause wrong results because the software takes this side sometimes also into account when trying to find the object. Due to this, the area of the object is bigger and the calculation of the center point of the object will be shifted more to the side of the effective object.

6.2.1 Correct Results of Position

The results can sometimes look almost perfect in the software, there can still be difficulties when the object needs to be picked up. The next images (Fig. 27) show the results of the software and the placement of the center point. The center point is visibly located perfect in the middle of the object, the exact location where the TCP of the robot needs to be. But the result of the location of the hand, this has a clear deviation.

This problem can be caused by just a temporary deviation but can also have to do with the calibration. Maybe by preforming a calibration with more dots can solve this problem. Also, maybe the deviation on the software while finding the

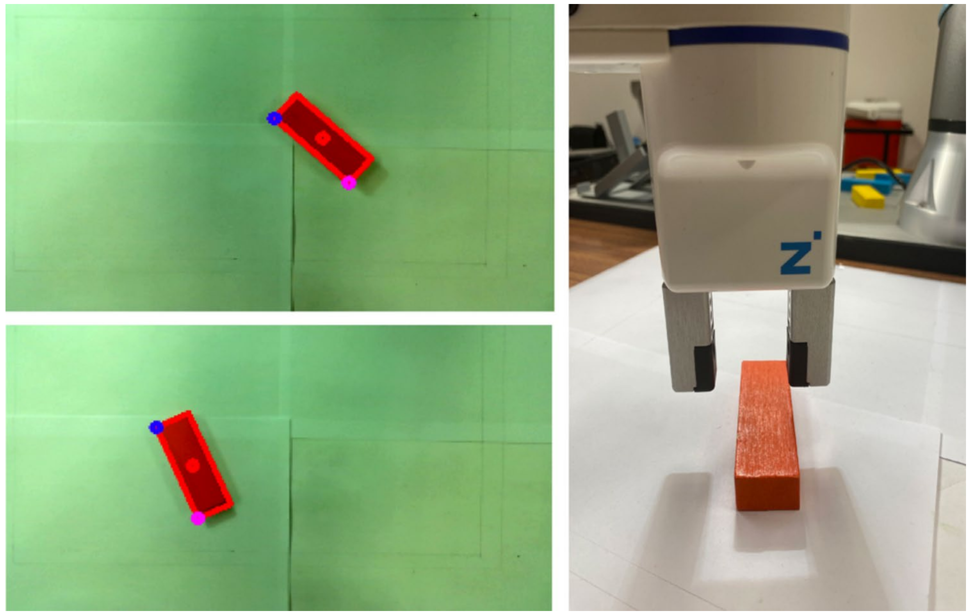


Fig. 27 Difference between software and robot location results

dots together with the deviation on putting the TCP manual on the dots or due to the reduced opening range of the gripper can be the reason, a sum of errors.

The results demonstrate that the proposed method outperforms conventional approaches in several aspects, including object recognition accuracy and robustness under varying lighting conditions. The integration of adaptive algorithms allows a rapid response to changes in the work environment, something that existing systems cannot achieve with the same effectiveness.

7 Conclusions

This paper presented an approach to solving the 'pick and place' problem using object detection with a collaborative robotic arm and a camera. Through experimentation and analysis, it has been demonstrated that the proposed approach can effectively eliminate the need for human intervention in such tasks, offering an efficient solution to the problem.

The use of advanced computer vision algorithms enabled the robot to recognize and manipulate objects with precision. While there is still room for improvement, particularly in location accuracy, this work provides a solid foundation for future developments.

Overall, the results of this study represent a significant application in the field of robotics and automation. Collaborative robots, as working tools, have the potential to execute pick and place tasks in all types of industries, including manufacturing.

The advantages and benefits of the method proposed in this paper, compared to general methods, include:

- Integration of a lighting source to improve image quality and object recognition rates.
- Addressing shadow issues by ensuring uniform illumination to enhance the reliability of object detection.
- Adaptation in dynamic environments through the use of adaptive algorithms and a low-cost vision system that allows for quick adjustments to changes in the work environment.
- Development and integration of adaptive algorithms and deep learning to improve real-time object recognition and classification.
- Integration of computer vision with deep learning to optimize both trajectory and object recognition, improving the precision of pick and place tasks.
- Flexibility of the proposed system that can be applied in different industrial scenarios, including the capacity to perform complex tasks such as quality inspection and the assembly of fragile parts.
- Testing and validation of the system in real industrial environments, ensuring the practical applicability of the proposed solutions.
- Integration of additional sensors and the use of an RGBD camera to improve the robot's spatial and depth perception.
- Use of deep learning for task control and simultaneous optimization of real-time object recognition and classification.
- Combination of AI and computer vision to automate complex tasks, such as quality inspection and assembly of parts in industrial environments.

Future work in this area should focus on developing advanced algorithms that account for variations in the shape and size of objects being picked up. Additionally, the recognition region can be expanded to accommodate new objects introduced to the pick and place task. This can also give the opportunity to integrate deep learning techniques into the application to deal with the change in shape and size.

Another area of interest is the use of the camera for identifying the drop-off point for the object. By using the camera, the robot can accurately place objects without the need for any external sensors or markers.

As future work, the authors propose the integration of more advanced neural networks and additional sensors to enhance the adaptability and precision of collaborative robotic systems in complex industrial environments.

Finally, the robot needs to be capable of picking up objects located at different heights. This can be done by using the depth vision of the camera.

Future work on pick and place applications with object recognition using a camera on a robotic hand should focus on developing innovative algorithms that consider multiple factors to enhance efficiency and accuracy.

The advantages of the proposed method, such as the ability to adapt to different industrial scenarios and the improvement in object recognition accuracy, are innovative compared to existing methods. This research demonstrated that the integration of vision systems and adaptive algorithms can further improve the integration of cobots in industry.

These benefits make the system more effective and capable of being applied to various types of more complex industrial applications.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Adriano A. Santos, Cas Schreurs, António Ferreira da Silva, Filipe Pereira, Carlos Felgueiras, António Lopes and José Machado. The first draft of the manuscript was written by Adriano A. Santos and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). This work was financially supported by Base Funding–UIDB/50022/2020 (LAETA) of INEGI–Institute of Science and Innovation in Mechanical and Industrial Engineering, Portugal, and UIDB/04730/2020 of CIETI–Center for Innovation in Industrial Engineering and Technology, funded by national funds through the FCT/MCTES (PIDDAC), Portugal.

Declarations

Ethics Approval Not applicable.

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Patil, S., Vasu, V., Srinadh, K.V.S.: Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends. Discov. Mech. Eng. **2**, 13 (2023). https://doi.org/10.1007/s44245-023-00021-8
2. Bormann, R., de Brito, B.F., Lindermayr, J., Omainska, M., Patel, M. Towards Automated Order Picking Robots for Warehouses and Retail. In: Tzovaras, D., Giakoumis, D., Vincze, M., Argyros, A. (eds).: Computer Vision Systems. ICVS 2019. Lecture Notes in Computer Science, 11754, 185–198. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34995-0_18.
3. Silveira, M., Santos, A.A., Pereira, F., da FerreiraSilva, A., FelgueirasRamosMachado, C.A.J.: 3D Vision Object Identification Using Yolov8. Int. J. Mechatron. Appl. Mech. **17**, 7–15 (2024). https://doi.org/10.17683/ijomam/issue17.1
4. Keshvarparast, A., Battini, D., Battaia, O., et al.: Collaborative robots in manufacturing and assembly systems: literature review and future research agenda. J. Intell. Manuf. **35**, 2065–2118 (2024). https://doi.org/10.1007/s10845-023-02137-w
5. Javaid, M., Haleem, A., Singh, R.P., Rab, S., Suman, R.: Significant applications of Cobots in the field of manufacturing. Cogn. Robot. **2**, 222–233 (2022). (**ISSN 2667-2413**)
6. Borboni, A., Reddy, K.V.V., Elamvazuthi, I., AL-QuraishiNatarajanAzhar Ali, M.S.E.S.S.: The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works. Machines **11**, 111 (2023). https://doi.org/10.3390/machines11010111
7. Kakade, S., Patle, B., Umbarkar, A.: Applications of collaborative robots in agile manufacturing: a review. Robot. Syst. Appl. **3**(1), 59–83 (2023). https://doi.org/10.21595/rsa.2023.23238
8. Asaad, H., Askar, S., Kakamin, A., Nayla, F.A.: Exploring the Impact of Artificial Intelligence on Humanrobot Cooperation in the Context of Industry 4.0. Appl. Comput. Sci. **20**, 138–155 (2024). https://doi.org/10.35784/acs-2024-21
9. Montini, E., Daniele, F., Agbomemewa, L., et al.: Collaborative Robotics: A Survey From Literature and Practitioners Perspectives. J. Intell. Robot. Syst. **110**, 117 (2024). https://doi.org/10.1007/s10846-024-02141-z
10. Jørgensen, T.B., Jensen, S.H.N., Aanæs, H., et al.: An Adaptive Robotic System for Doing Pick and Place Operations with Deformable Objects. J. Intell. Robot. Syst. **94**, 81–100 (2019). https://doi.org/10.1007/s10846-018-0958-6
11. Sezer, A., Altan, A.: Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. Soldering Surface Mount Technol. **33**(5), 291–298 (2021). https://doi.org/10.1108/SSMT-04-2021-0013
12. Yağ, İ, Altan, A.: Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments. Biology (Basel) **11**(12), 1732 (2022). https://doi.org/10.3390/biology11121732
13. Anjum, M.U., Khan, U.S., Qureshi, W.S., Hamza, A., Khan. W.A.: Vision-Based Hybrid Detection For Pick And Place Application

In Robotic Manipulators. 2023 International Conference on Robotics and Automation in Industry (ICRAI), Peshawar, Pakistan, 1–5 (2023). https://doi.org/10.1109/ICRAI57502.2023.10089602.
14. Borrell, J., Perez-Vidal, C., Segura, J.V.: Optimization of the pick-and-place sequence of a bimanual collaborative robot in an industrial production line. Int. J. Adv. Manuf. Technol. **130**, 4221–4234 (2024). https://doi.org/10.1007/s00170-023-12922-9
15. Gomes, N.M., Martins, F.N., Lima, J., Wörtche, H.: Reinforcement Learning for Collaborative Robots Pick-and-Place Applications: A Case Study. Automation **3**, 223–241 (2022). https://doi.org/10.3390/automation3010011
16 Javaid, M., Haleem, A., Singh, R.P., Rab, S., Suman, R.: Significant applications of Cobots in the field of manufacturing. Cogn. Robot. **2**, 222–233 (2022). https://doi.org/10.1016/j.cogr.2022.10.001. (**ISSN 2667-2413**)
17. Pereira, F., Pinto, L., Soares, F., Vasconcelos, R., Machado, J., Carvalho, V.: Online yarn hairiness– Loop & protruding fibers dataset. Data Brief **54**, 110355 (2024). https://doi.org/10.1016/j.dib.2024.110355. (**ISSN 2352-3409**)
18 Patil, S., Vasu, V., Srinadh, K.V.S.: Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends. Discov. Mech. Eng. **2**, 13 (2023). https://doi.org/10.1007/s44245-023-00021-8
19. Universal Robots. https://www.universal-robots.com/about-universal-robots (2024). Accessed on 18 April 2024.
20. Zimmer group. HRC-03 2-jaw parallel gripper. https://www.zimmer-group.com/en-us/technologies-components/components/handling-technology/grippers/hrc/collaborative/2-jaw-parallel-grippers/hrc-03 (2024). Accessed on 18 April 2024.
21. Intel RealSense. https://www.intelrealsense.com/depth-camera-d435i (2024). Accessed on 18 April 2024.
22. Neadvance. Available online: https://neadvance.com (2024). Accessed on 18 April 2024.
23. Niop. https://neadvance.com/products/niop (2024). Accessed on 18 April 2024.
24. Szeliski, Richard. Introduction. In Computer vision: algorithms and applications, 2nd ed., Springer Nature, Seattle, WA, USA, pp. 1–26 (2022).
25. P. Ongsulee. Artificial intelligence, machine learning and deep learning. In 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand (2017). https://doi.org/10.1109/ICTKE.2017.8259629.
26. Dziubek, M., Rysiński, J., Jancarczyk, D.: Exploring the ViDiDetect Tool for Automated Defect Detection in Manufacturing with Machine Vision. Appl. Sci. **13**, 11098 (2023). https://doi.org/10.3390/app131911098
27 Haffner, O., Kučera, E., Rosinová, D.: Applications of Machine Learning and Computer Vision in Industry 4.0. Appl. Sci. **14**, 2431 (2024). https://doi.org/10.3390/app14062431
28. Cognex, Deep Learning for Factory Automation. Available online: https://www.cognex.com/what-is/deep-learning/for-factory-automation (2024). Accessed on 11 April 2024.
29. Pérez, L., Rodríguez, Í., Rodríguez, N., Usamentiaga, R., García, D.F.: Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. Sensors **16**, 335 (2016). https://doi.org/10.3390/s16030335
30. Jahr, I.: Lighting in machine vision. In Handbook of Machine and Computer Vision: The Guide for Developers and Users, pp. 63–178. Wiley-VCH Verlag GmbH & Co. KGaA (2017). https://doi.org/10.1002/9783527413409.ch3
31. Riba, J.-R.: Application of Image Sensors to Detect and Locate Electrical Discharges: A Review. Sensors **22**, 5886 (2022). https://doi.org/10.3390/s22155886
32 Garnir, H.P., Lefèbvre, P.H.: Quantum efficiency of back-illuminated CCD detectors in the VUV region (30–200 nm). Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. Atoms **235**, 530–534 (2005). https://doi.org/10.1016/j.nimb.2005.03.238
33. Katikala, H.B., Pitchaiah, T., Murthy, G.R.: Low Readout Noise Photodiode based CMOS Image Sensor with High Fill Factor for Biomedical application. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India (2022). https://doi.org/10.1109/DELCON54057.2022.9753074
34. Ansari, S.: Core Concepts of Image and Video Processing. In Building Computer Vision Applications Using Artificial Neural Networks, Apress, Berkeley, CA, pp. 15–33 (2023). https://doi.org/10.1007/978-1-4842-5887-3_2
35 Ma, D., Dang, B., Li, S., Zang, H., Dong, X.: Implementation of computer vision technology based on artificial intelligence for medical image analysis. Int. J. Comput. Sci. Inf. Technol. **1**(1), 69–76 (2023). https://doi.org/10.62051/ijcsit.v1n1.10
36. Distante, A., et al.: Handbook of image processing and computer vision, From Energy to Image, Vol. 1. Springer International Publishing, AG (2020). https://doi.org/10.1007/978-3-030-38148-6
37. Distante, A., et al.: Handbook of image processing and computer vision, From Image to Pattern, Vol 2. Springer International Publishing, AG (2020). https://doi.org/10.1007/978-3-030-42374-2.
38. Distante, A., et al.: Handbook of image processing and computer vision, From Pattern to Object, Vol. 3. Springer International Publishing, AG, (2020). https://doi.org/10.1007/978-3-030-42374-2
39. Yanowitz, S.D., Bruckstein, A.M.: A new method for image segmentation. Comput. Vision Graph. Image Process. **46**(1), 82–95 (1989). https://doi.org/10.1016/S0734-189X(89)80017-9
40. Hadwiger, H.: Vorlesungen Über Inhalt, Oberfläche und Isoperimetrie. Springer Berlin, Heidelberg (1957). https://doi.org/10.1007/978-3-642-94702-5.
41. Minkowski, H.: Volumen und Oberfläche. Math. Ann. **57**(4), 447–495 (1903). https://doi.org/10.1007/BF01445180
42. Dawson-Howe, K.: In A Practical Introduction to Computer Vision with OpenCV, 1st ed, pp. 1–65. John Wiley & Sons (2014).
43 Sivkov, S., Novikov, L., Romanova, G., Romanova, A., Vaganov, D., Valitov enVasiliev, M.S.: The algorithm development for operation of a computer vision system via the OpenCV library. Procedia Computer Science **169**, 662–667 (2020). https://doi.org/10.1016/j.procs.2020.02.193

Adriano A. Santos has the PhD degree in Mechanical Engineering, from University of Porto, Portugal. He is Director of the Postgraduate Program in Industrial Automation and Robotics and Associate Professor at Mechanical Engineering Department of School of Engineering, Polytechnic of Porto. Is a member of the research group Intelligent Systems & Control at LAETA-INEGI and CIDEM at ISEP. He is the author of numerous national and international publications, including journal articles, Books and book chapters, and reviews of international papers. His main interests are related to Industry 4.0, more specifically in the design of the mechatronic (pneumatic and oil-hydraulics) system, Maintenance, Robotics, AI and Mechanical engineering.

Cas Schreurs studied Industrial Engineer at the University of Leuven, Belgium and finished his studies with a master thesis at Mechanical Department at Polytechnic of Porto. The thesis further developed his enthusiasm for programming and robotics. Due to his specific interest in automation, programming, and optimization, he chose to begin his career as a PLC engineer at Movu-Robotics. In this role, he focuses on developing automated systems for warehouses of all sizes.

António Ferreira da Silva received his PhD degree in Mechanical Engineering, from University of Porto, Portugal. He is Director of Automation Lab. and Associate Professor at Mechanical Engineering Department of School of Engineering, Polytechnic of Porto. Is a member of the Institute of Science and Innovation in Mechanical Engineering and Industrial Engineering (INEGI) and CIDEM. He is the author of numerous national and international publications, including journal articles, Books and book chapters, and reviews of international papers. His main interests are related to mechatronic (pneumatic and oil-hydraulics) systems, Robotics, AI and Mechanical engineering.

Filipe Pereira is a specialist in Electronics, Automation, Robotics, and Industrial Control with 24 years of experience. He holds a PhD in Electronic and Computer Engineering and a Master's in Electronic Engineering and Computer Science, both with a focus on Automation and Industrial Control. Filipe is a member of the MEtRICs and Algoritmi Research Centers at the University of Minho and INEGI. He has authored numerous national and international publications, including journal articles and book chapters, and reviews international papers. As a certified trainer in Automation, Robotics, and Industry 4.0, he collaborates with leading companies. Filipe is also vice-president of the Electronics and Telecommunications college at OET. His interests include Industry 4.0, Robotics, AI, and Electronics.

Carlos Felgueiras achieved a PhD in Electrical and Computer Engineering from the Faculty of Engineering-University of Porto, Portugal in 2008. His PhD addressed the development of mechanisms to support testing and debugging operations in mixed-signals circuits. Professional career activities included: testing of products, developing and maintenance of electronic equipment, training teacher at Texas Instruments plant. His teaching activity started in 1993 and is Adjunct Professor at the Department of Electrical Engineering at ISEP- P.Porto, and is Integrated Researcher at CIETI since 2001. He was Deputy Director of the Electrical Department between and Director of the master's degree in Sustainable Energies between. His areas of research include Design for Debugging and Testing Mixed Signal Circuits, Remote Experimentation, e-learning, Renewable Energy Sources and Smart Buildings.

António M. Lopes has the Ph.D. and Habilitation in mechanical engineering from the Faculty of Engineering of the University of Porto (FEUP). He is with the Associated Laboratory for Energy, Transport, and Aeronautics (LAETA), Institute of Science and Innovation in Mechanical Engineering and Industrial Engineering (INEGI). He is the coordinator of the research group Intelligent Systems & Control at LAETA-INEGI. His research areas are Automation, Robotics, Complex systems, and Fractional-order systems. He coauthored more than 230 SCI journal papers. He is coeditor-in-chief of the Journal of Environmental Accounting and Management, and Journal of Machine Design and Automation Intelligence, and Section editor-in-chief of Fractal and Fractional. He is currently an Associate Professor of Mechanical Engineering at FEUP.

José Machado concluded Habilitation Title in February 2019 at University of Minho, Portugal. He received his PhD degree in Mechanical Engineering, from University of Minho, Portugal and, in simultaneous, from Ecole Normale Superieure de Cachan, France, in 2006. He is Deputy Director of MEtRICs Research Center and Associate Professor w/ Habilitation at Mechanical Engineering Department of University of Minho. He has co-authored more than 250 refereed journal and conference proceedings papers. He coordinates several Research and Technology Transfer Projects on Mechatronics and Automation domains. His main interests are related with Industry 4.0, more specifically, on the design and development of Cyber-Physical Systems and mechatronic systems design with special focus on manufacturing industry and biomedical applications, wellbeing and/or rehabilitation.