POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Robotic assembly in an unstructured environment

Tutor

Prof. ANTONELLI Dario

Candidate

Azizjon Khudayberganov 300807

April 2024

Abstract

The objective of this thesis is to devise a computer vision algorithm tailored for use by a collaborative robot (cobot) to identify various mechanical components within a workspace. This capability is crucial for enabling robotic assembly tasks in unstructured environments. A successful assembly operation requires the robot to retrieve components from the workspace in a precise sequence to execute tasks accurately. Therefore, the primary aim of the computer vision algorithm is to streamline this process, allowing the cobot to select the appropriate mechanical component even in scenarios where the component's position is unknown or variable within the workspace. In typical assembly scenarios, mechanical components may share similar characteristics, such as shape or material, but differ in key technological features, like size. Hence, the computer vision algorithm must discern these distinctions to differentiate the relevant mechanical components from others in the workspace effectively. The thesis is structured into two main segments: The first part focuses on developing the algorithm to identify the correct mechanical component among several present in the workspace for the cobot to manipulate. This was achieved by utilizing the integrated camera of the TM5-900 cobot. In the second part, following the recognition of the correct mechanical component, the cobot approaches it and determines its precise barycenter position to facilitate grasping, while considering moments of inertia. This aspect of the work was accomplished using TMFlow software associated with an Omron cobot, which utilizes the cobot's integrated camera. Subsequently, after correctly grasping the component, the robot can manipulate it for assembly purposes.

INDEX

Introduction

Objective

The objective of this thesis is to create a computer vision algorithm tailored for implementation in a collaborative robot (cobot). This algorithm aims to enable the cobot to accurately identify various mechanical components within its operating environment and assemble them. The recognition of different and similar mechanical pieces recognized by a cobot can be used for robotic assembly purposes in non-structured environments. A robotic assembly operation of a mechanical part requires that the robot take the pieces present in the specified range of workspace in an appropriate order to correctly execute the task step by step. Therefore, the goal of the computer vision algorithm is to automate all the processes: the cobot can take the right mechanical piece without knowing the exact coordination of the pieces and also, they can be moved in the range of the camera vision. In a general assembly operation, the mechanical pieces seem very similar to each other, for instance, they can have the same shape, or they can have been made of the same material or the same dimension or the same number of holes but there is at least one unique difference in technological feature. For this reason, the computer vision algorithm must differentiate the pieces from the others in the workspace, using every kind of technological feature. [1] Unstructured environments contain elements or variables that unpredictable obstacles, and a traditional robot cannot leverage pre-established programming to reliably complete a task. As a result, there is a current requirement for a robot capable of comprehending its surroundings and dynamically generating a path to accomplish tasks. This necessitates a significant shift in robot programming methodologies and the selection of sensors to enable the robot to perceive and interpret its environment effectively. The work begins with the first part that the integrated camera must recognize what is the correct piece among several mechanical pieces present in the workspace, that the cobot must grab. According to, specific small differences camera can recognize an appropriate part of the assembly. The next step is after recognizing the piece, the cobot looks for the correct center and frame and moves to the right above the piece to pick the object. This process can be done with the Image recognition feature of the TMFlow software of Omron cobot. After grabbing the piece, the cobot can continue the collaborative assembly process with human.

1. Collaborative Robots in Industry 4.0

1.1 Robots and cobots in assembly

The name “cobot” means “collaborative robot”. As the name says it is intended for direct human-robot interaction under the same workspace [9]. Apart from traditional industrial robots, cobots' safety may involve either utilizing lightweight materials with rounded edges and inherent speed and force limitations or employing sensors and software to ensure safe operation. There are essentially two main types of robots: industrial robots are used in automation and service robots utilized for household and professional purposes. Service robots, which are intended to work alongside humans, can be considered as cobots. Unlike traditional industrial robots that typically operate separately from humans, often behind fences or protective barriers, cobots eliminate this physical separation. [8] In 1996, Northwestern University professors J. Edward Colgate and Michael Peshkin invented cobots. Their patent in the United States, titled "Cobots", describes "a device and method for direct physical interaction between a person and a general-purpose manipulator controlled by a computer." The inception of this invention can be traced back to a 1994 General Motors initiative spearheaded by Prasad Akella of the GM Robotics Center and further supported by a 1995 research grant from the General Motors Foundation. The objective was to find a solution to ensure the safety of robots or robot-like equipment for collaborative work alongside humans. Industrial robots *(Figure 1)* prioritize efficiency over safety, often operating at high speeds, carrying heavier loads, and being larger compared to cobots. Consequently, they are typically segregated from human workers by cages or barriers. Cobots *(Figure 2)* [7], on the other hand, are smaller, handle lighter loads, move at slower speeds, and are designed with curved edges to prioritize worker safety. Moreover *(Figure 3)*, cobots can be swiftly set up, while industrial robots tend to be more intricate to program, usually requiring expertise in specific programming languages. Cobots are also easily adaptable to various tasks, offering greater flexibility. In contrast, industrial robots are engineered for specific functions, and it can be time-consuming to reconfigure them.

Figure 1: *Industrial cobot*

Figure 2: *Omron robot*

	Traditional Industrial Robots	Omron Cobots
Safety	Needs a physical barrier, such as a fence or cage, to ensure safety.	Designed to be inherently safe but may need safety sensors to ensure that the application is safe (e.g. Omron safety laser scanner) based on risk assessment. Typically does not need physical barrier if working in collaborative mode. Software safety setting is easy with graphical user interface.
Workspace	Separated from human workspace.	Can be shared with people.
Footprint	Large	Small
Flexibility	No. Fixed to one location and works on dedicated task.	Yes. Can be moved between locations during the day to work on different tasks. Built-in camera and Landmark positioning enable quick relocation.
Programming	Difficult. Requires skill and training.	Easy. Can be done with minimal training.
Setup	Requires advanced skills and is time-consuming.	Quick and easy.
Application	Fit for mass production at high speeds.	Fit for high-mix, low-volume production at a speed comparable to human workers. Can be used at high speeds with safety measures.
Cycle Time (Pick & Place)	Down to seconds	Over 5 seconds
Speed of Process (Path)	Below 8.2 m/s	Below 1.4 m/s
Repeatability	+/- 0.02 mm	+/- 0.05 mm
Environment	IP requirements above IP54	IP54 (robot arm), IP32 (control box)
Process Complexity	Can be complex	Should be simple

Figure 3: *Table*

1.2 Robots in an unstructured environment

In structured environments, such as manufacturing plants with robotic arms performing repetitive tasks or robotic operations like dozing, excavation, and haulage in mining, the conditions are typically stable and predictable. In these settings, there are few variables, that allow robots to effectively execute pre-programmed tasks without encountering significant environmental changes. On the other hand, in unstructured environments, the work conditions tend to be unpredictable, characterized by elements or variables that defy anticipation. Traditional robots are unable to rely on pre-established programming to consistently accomplish tasks in such environments. In unstructured environments [3], robots typically have limited awareness of their surroundings, with objects capable of changing states without the robot's knowledge. Manipulation tasks may demand the end effector to follow specific trajectories rather than merely reaching predetermined positions. These challenges significantly complicate motion generation. Explicit coordination between planning and sensing becomes essential to manage dynamic environments, which in turn expands the complexity of the state space. [4] Additionally, more intricate task demands necessitate high-frequency feedback, imposing stringent requirements. However, the majority of real-world environments exhibit considerable structure. Buildings are typically organized into hallways, rooms, and doors, while outdoor spaces feature paths, streets, and intersections. Common objects like shelves, boxes, tables, and chairs often have preferred approach directions. This pertinent information is disregarded when planners solely operate in configuration space. Consequently, most motion planners must assume that the environment is fully understood and remains unchanged during the planning process. Robot perception has been the focus of extensive research over several decades. However, much of the work in this field relies on assumptions that do not hold in unstructured and dynamic environments. For instance, in face recognition, assumptions are often made regarding the position and orientation of the person in the image. Similarly, object segmentation results are typically based on the ability to differentiate between objects and backgrounds using color variations, while object recognition often involves comparing similarities to a predefined set of objects. In unstructured environments, however, controlling position and orientation becomes challenging, assumptions about colors and shades are hard to justify, and the range of potential objects the robot may encounter is vast and unpredictable. Manipulating objects in unstructured environments presents numerous challenges absent in structured settings. In such environments, crucial object properties necessary for manipulation cannot be predetermined. Information about objects must be gathered via sensors, which frequently yield ambiguous data, introduce uncertainty, and offer redundant information concerning the manipulation task. Additionally, manipulating objects in unstructured and dynamic environments often demands swift responses to rapidly evolving conditions. Recognizing the structural layout of the environment can aid in manipulation tasks. Yet, obtaining accurate information about the environment's state proves highly challenging in unstructured settings. Objects might be partially obscured, lighting could be inadequate, and the intended use of an object might be hard to discern. Such ambiguity in sensor data heightens uncertainty about the environment, consequently expanding the size of the state space. Integrating manipulation and perception closely can mitigate this complexity in the state space. Interaction with humans offers robots an additional avenue to simplify the challenges posed by unstructured environments. Humans can indicate noteworthy features through gestures, impart new skills through demonstration, or transfer knowledge using language. Furthermore, numerous real-world tasks necessitate collaboration between humans and robots. Collaboration between humans and robots in task execution involves communication regarding objects, tools, and objectives. Numerous tasks entail transferring objects between a human and a robotic partner. Instructing the robot verbally presents challenges due to the intricacy of both the environment and the robot's mechanisms. The robot must determine the optimal hand placement, orientation, finger configuration, and force application.

2. Characteristics of the cobot

2.1 Omron TM5-900

In the thesis, the cobot Omron TM5-900 *(Figure 4)* is used, TM5 is the model of the robot and it can reach up to 900mm. Since it is a collaborative robot [5], it was designed to guarantee safety during interaction between humans and cobots in the same work area. The OMRON TM Collaborative robots [7] are designed to perform various tasks and applications, ensuring flexible production with the help of flexible programming and easy mobility. A key advantage of these cobots is their ability to fit into compact spaces, making them adaptable to nearly any factory setting. The most significant benefit of this cobot lies in its integrated vision system with 5M pixels characteristics, tailored for pattern recognition, object positioning, and servoing. During this thesis, the integrated camera is utilized to identify the pattern of each similar piece and its center of mass for grasping purposes.

Figure 4: *Omron TM5-900*

2.2 Characteristics of TM5-900

The characteristics of the cobot are shown below in *Figure 5* [6][7].

<table>
<tr><td colspan="2">Product Name</td><td>TM5-700</td><td>TM5M-700</td><td>TM5M-700 SEMI</td><td>TM5-900</td><td>TM5M-900</td><td>TM5M-900 SEMI</td></tr>
<tr><td colspan="2">Part Number</td><td>RT6-000700X</td><td>RT6-010700X</td><td>RT6-010701X</td><td>RT6-000900X</td><td>RT6-010900X</td><td>RT6-010901X</td></tr>
<tr><td colspan="2">Weight (kg)</td><td colspan="3">22.1</td><td colspan="3">22.6</td></tr>
<tr><td colspan="2">Controller Weight (kg)</td><td>13.5</td><td>14.5</td><td>14.5</td><td>13.5</td><td>14.5</td><td>14.5</td></tr>
<tr><td colspan="2">Max Payload (kg)</td><td colspan="3">6</td><td colspan="3">4</td></tr>
<tr><td colspan="2">Reach (mm)</td><td colspan="3">700</td><td colspan="3">900</td></tr>
<tr><td colspan="2">Mounting</td><td colspan="6">Wall, Table, Ceiling</td></tr>
<tr><td colspan="2">Typical Speed (m/s)</td><td colspan="3">1.1</td><td colspan="3">1.4</td></tr>
<tr><td rowspan="4">Joint Range</td><td>Joint 1</td><td colspan="6">±270°</td></tr>
<tr><td>Joint 2, 4, 5</td><td colspan="6">±180°</td></tr>
<tr><td>Joint 3</td><td colspan="6">±155°</td></tr>
<tr><td>Joint 6</td><td colspan="6">±270°</td></tr>
<tr><td rowspan="2">Joint Speeds</td><td>Joint 1, 2, 3</td><td colspan="6">180°/s</td></tr>
<tr><td>Joint 4, 5, 6</td><td colspan="6">225°/s</td></tr>
<tr><td colspan="2">Repeatability (mm)</td><td colspan="6">±0.05</td></tr>
<tr><td colspan="2">IP</td><td colspan="6">IP54 (robot arm), IP32 (control box), IP40 (robot stick)</td></tr>
<tr><td colspan="2">Cleanroom Class</td><td colspan="6">ISO Class 5</td></tr>
</table>

Figure 5: *Characteristics of TM5-900*

2.3 Omron TM5-900 components

As shown in *Figure 6*, the cobot has a base, six joints, and an end-effector.

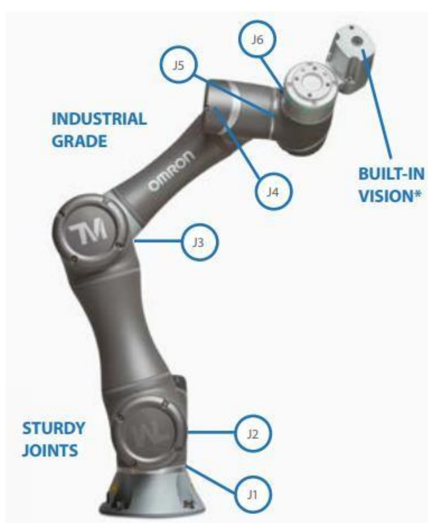

Figure 6: *Joints of the cobot*

In *Figure 7* there is an end-effector with three buttons that focus on a variety of functions. The number 1 in *Figure 7* is the vision button which is used to add a new vision node to the execution flowchart in the program since the cobot is programmed via a software called TM flow. The number 2 in *Figure 7* is the point button used to add a new recorded point (current position) of the cobot in the flowchart of the execution of the cobot program. The number 3 is the free button, by pressing it the cobot can be manually hand-guided, which means that the worker can move the cobot by hand in servo-assisted mode. This is important because the users can guide the cobot into positions freely and automatically the cobot records the position in the software and this makes it easy to set points to the cobot. *Figure 8* shows the other three elements that are present in the end-effector. In the Analog I/O Connector *(the number 4 in Figure 8)* there are five pins: the supply, the reference, the digital input of the type NPN, the digital output of the type NPN and the last one is the analog input ± 10 V. The indicator light ring *(the number 5 in Figure 8)* is important since shows the robot status since the robot can be in manual mode(green light), automatic mode (blue light). Moreover, if the indicator light ring is red is because there is an error or the cobot is in initialization mode. When the light is sky blue, it means that the cobot is in a safe startup mode. In the Digital I/O Connector *(the number 6 in Figure 8)*, instead, there are eight pins: the supply, the reference, three digital inputs of the type NPN, and three digital outputs of the type NPN. As shown in *Figure 9*, the last three elements that are present in the hand of the cobot are the built-in vision system, the gripper button, and the end-of-arm tooling flange. The built-in vision system *(the number 7 in Figure 9)*, thanks to the hand guiding and the landmark positioning, allows the cobot to do quick setup for the pick-and-place tasks. The gripper button *(the number 8 in Figure 9)*, instead, is a way in programming to close and open the gripper and adds these operations to the execution flow of the program of the cobot. Under the end-of-arm tooling flange *(the number 9 in Figure 9)* it can be possible to insert different tools. For the thesis is inserted a gripper which is the RobotiQ 2F-85 model.

Figure 7

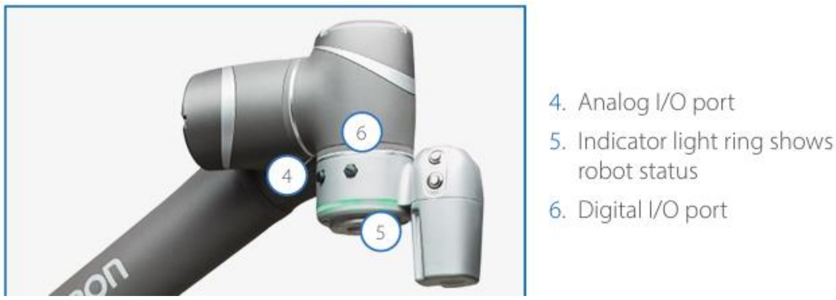

Figure 8

Figure 9

2.4 Robotiq 2F-85 gripper

To perform the program perfectly, pick and place parts there is installed an adaptive gripper Robotiq 2F-85 *(Figure 10)*. 2F-85 name of the model means two-finger gripper with an 85mm opening *(Figure 11)*. The 2-Finger Gripper has two articulated fingers that each have two joints (two phalanxes per finger), as shown in the figure below. The grasp-type gripper can engage up to five points of contact with an object (two on each of the phalanges plus the palm). The fingers are under-actuated, meaning they have fewer motors than the total number of joints. This configuration helps the fingers to automatically adapt to the shape of the object they grasp which simplifies the control of the gripper [11].

Figure 10: *Robotiq 2F-85*

Specification	2-FINGER 85		2-FINGER 140	
	Metric Units	Imperial Units	Metric Units	Imperial Units
Stroke	85 mm	3.35 in	140 mm	5.5 in
Minimum object diameter (for encompassing)	43 mm	1.69 in	90 mm	3.5 in
Maximum height	162.8 mm	6.4 in	232.8 mm	9.15 mm
Maximum width	148.6 mm	5.85 in	202.1 mm	8.0 in
Weight	925 g	2.04 lbs	1,025 g	2.25 lbs
Grasp Force	20 to 235 N	4.5 to 52.8 lbf	10 to 125 N	2.2 to 28.1 lbf
Finger speed	20 to 150 mm/s	0.8 to 5.9 in/s	30 to 250 mm/s	1.2 to 9.8 in/s
Position repeatability[1]	0.05 mm	0.002 in	0.08 mm	0.003 in
Force repeatability	+/- 10%			
Position resolution[2]	0.4 mm	0.016 in	0.6 mm	0.022 in
Grasp force resolution	Maximum force calculation below; refer to the **Force Control** section			

Figure 11: *Characteristics of 2F-85 and 2F-140*

2.5 Connection architecture of the system

The entire system architecture *(as illustrated in Figure 12)* comprises the collaborative robot, the control box, and the TMflow software tool [5]. The control box is linked to the cobot and is responsible for its operation. The robot stick, also connected to the control box, allows basic operations on the cobot through its buttons, such as power control, execution, speed adjustment, and pause or stop commands. Moreover, with the TMflow software installed on the PC connected to the control box, it becomes possible to develop and execute different programs and tasks for the cobot.

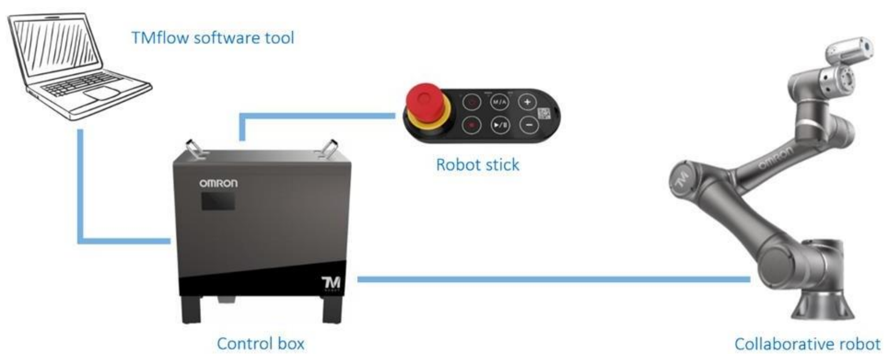

Figure 12: *Connection of the TM5-900*

2.6 Integrated camera

The Omron TM5-900 includes also an integrated camera (*Figure 13*) on the wrist. Its camera is exploited in the second part of the work of the thesis: the cobot must measure the exact position of the barycenter to grasp it by paying attention to the moments of inertia to manipulate the piece correctly [5]. This camera system is engineered to deliver high-resolution imagery and real time processing, facilitating the seamless detection and identification of objects within its operational ambit.

Distinctive Features of the Omron TM5-900's Integrated Camera:

- High-Resolution Imaging: The integrated camera embedded within the Omron TM5-900 offers superior imaging resolution, enabling meticulous examination of objects within its visual scope. This heightened resolution ensures precise detection, even with diminutive or intricate objects.

- Real-Time Processing: Equipped with formidable onboard processing prowess, the TM5-900 is adept at executing object detection tasks in real-time. This expeditious decision-making capability empowers the robot to dynamically respond to environmental alterations.

- Versatile Object Detection: The camera system integrated into the TM5-900 is adaptable to a myriad of object detection tasks, spanning from identifying objects on conveyors to facilitating quality control assessments and negotiating obstacles within its path. Its versatility renders reliable performance across diverse operational contexts.
- Customizable Algorithms: Users possess the flexibility to tailor detection algorithms to meet specific application requisites. This customization empowers fine-tuning detection parameters, thereby optimizing system performance across varying scenarios and environments.
- Integration with Robotic Workflows: Seamlessly integrating with the TM5-900's robotic workflow, the integrated camera empowers the robot to undertake intricate tasks necessitating object detection. This integration streamlines automation processes and amplifies the robot's proficiency in activities such as pick-and-place operations, sorting tasks, and assembly endeavors. The characteristics of the camera are shown in *Figure 14*.

Figure 13: *Integrated camera*

Resolution	5Mpx
Autofocus	100mm -> +∞ (rolling shutter)
View angle	60° diagonal
Sensor dimension	¼ inch

Figure 14: *Table*

3. Object detection with features

3.1 Object detection and computer vision

Object detection is a task in computer vision that seeks to identify objects within digital images. It's a form of artificial intelligence that trains computers to perceive visual data similar to humans, focusing on recognizing and categorizing objects based on their semantic attributes.

Object localization involves determining the precise location of objects within an image using bounding boxes *(Figure 15)*:

- Input: An image with one or more objects, like a photograph.
- Output: One or more bounding boxes (e.g., defined by coordinates, width, and height).

Object classification assigns detected objects to specific categories. Object detection combines these subtasks of localization and classification to estimate both the position and type of object instances in one or more images simultaneously:

- Input: An image with one or more objects, such as a photograph.
- Output: One or more bounding boxes (e.g., defined by coordinates, width, and height), and a class label for each bounding box [1][2].

The main goal of this thesis is to distinguish the correct mechanical parts among similar ones because this thesis is related to object detection tasks.

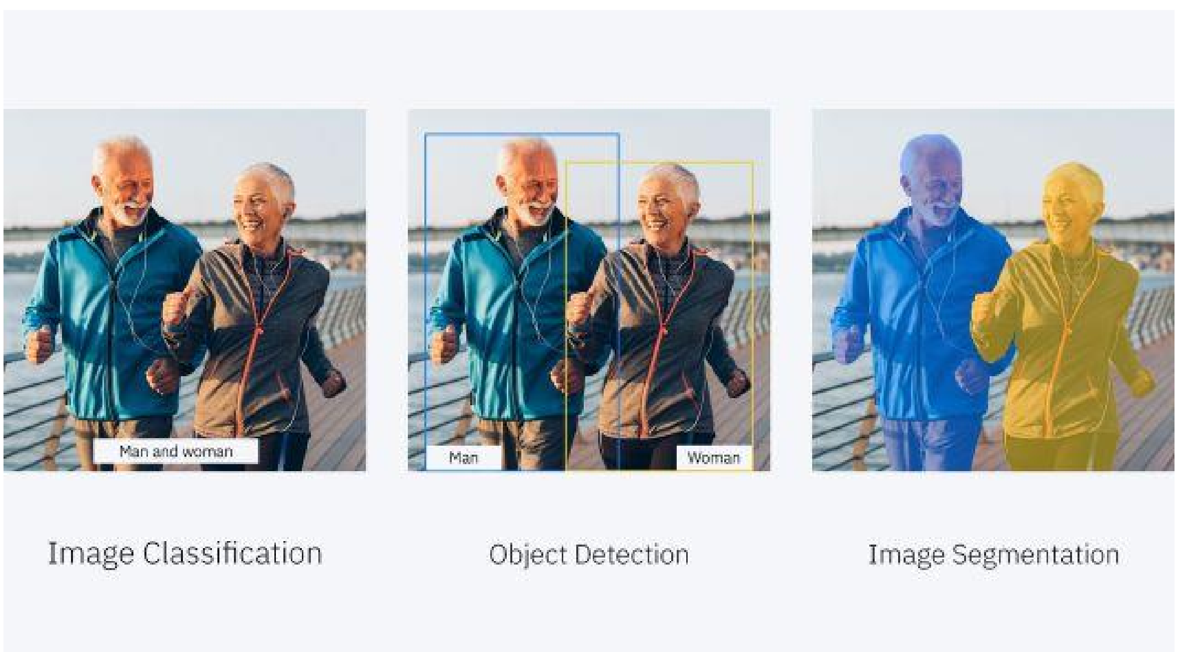

Figure 15: *Object detection [1]*

3.2 Object detection using the integrated camera of Omron TM5-900

The utilization of the Omron TM5-900 robotic platform introduces cutting-edge object detection capabilities through its integrated camera system, which serves as a cornerstone in numerous industrial automation scenarios. The process of object detection with the Omron TM5-900 involves several steps and considerations. Here's a generalized outline of the process:

- Camera Calibration:

Before object detection can begin, it's essential to calibrate the camera of the Omron TM5-900. This calibration ensures accurate measurements and image analysis.

- Image Acquisition:

The integrated camera captures images of the robot's surroundings or the area of interest where object detection is required. These images serve as input data for the object detection algorithms In Figures 16 and 17 there are shown parameter settings for clear recognition assembling parts. These parameter settings are chosen after a number of tests to obtain a clear view of an object and distinguish the object's frame from the background. They are chosen as an optimal value for this assembly project. Configuring parameters depend on the characteristics of an object: size, shape, edges, holes, color and others. During recognition of the screws, as the screws' color is similar with background a lighting of the integrated camera is turned on for a maximum value 255, while detecting the main part the lighting parameter adjusted to 8. Apart from the parameter settings, there are also vision enhancing parameters: Contrast enhancement, color plane extraction, smoothing, thresholding and morphology. For screw detection, the contrast enhancement is used with parameters: contrast – 2.99, brightness – 89.38, gamma – 1.00 and color plane is RGB.

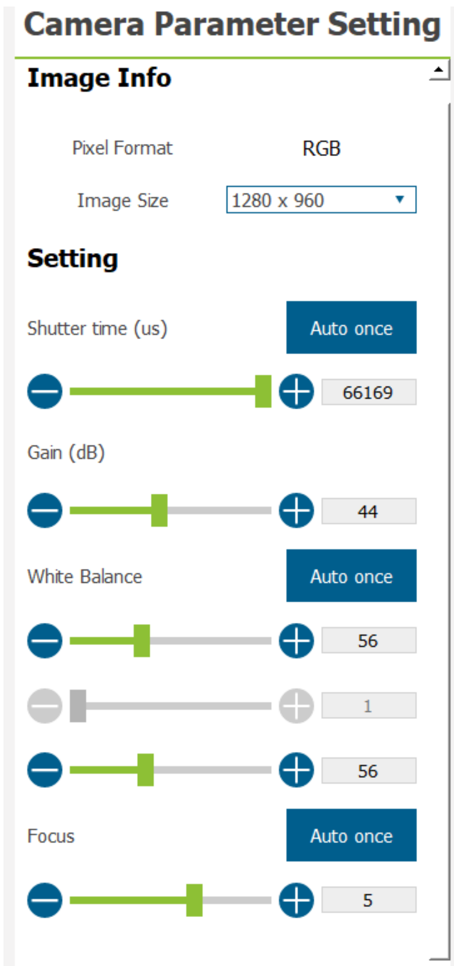

Figure 16

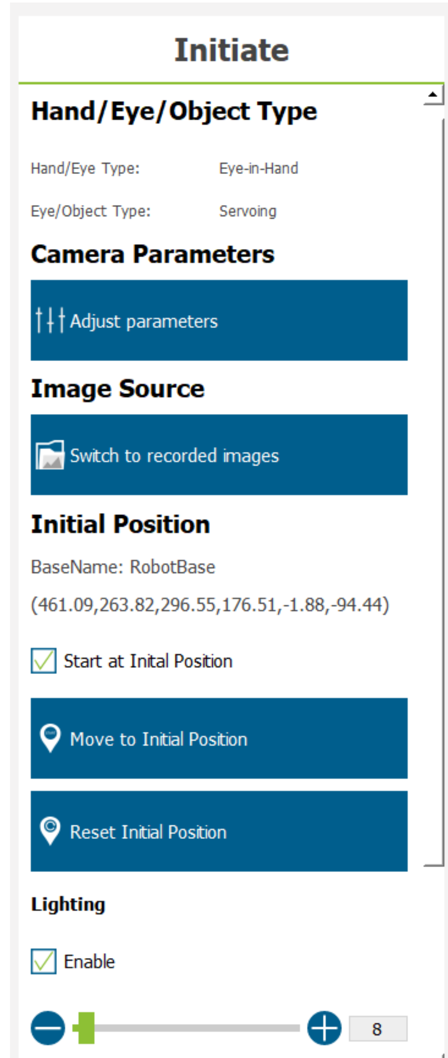

Figure 17

- Preprocessing:

Preprocessing techniques may be applied to the captured images to enhance their quality and improve the performance of the object detection algorithms. This could include operations such as noise reduction, image denoising, contrast enhancement, or image normalization.

- Object Detection Algorithm:

The Omron TM5-900 utilizes advanced object detection algorithms to analyze the captured images and identify objects within them. These algorithms may include techniques such as edge detection, feature extraction, template matching, machine learning, or deep learning approaches like convolutional neural networks (CNNs).

- Feature Extraction:

Once objects are detected within the images, relevant features of these objects are extracted. These features could include shape, size, color, texture, or any other distinguishing characteristics that help differentiate one object from another. Figures 18 and 19 the process of how to find a part with the detection of edges and there are parameters to adjust the searching range or how to exactly recognize an object and how many objects. For all parts, there are different pattern matching parameters as they are different from each other. There is main 3 parameters for better accuracy and precision, the first one a number of Pyramid Layers. Each layer is a subsequent blur and subsample of the original image. The bigger the number of Pyramid Layers, the most efficient will be the algorithm. The finest details, the less the number of Pyramid Layers. The next is minimum score, each layer has matching coefficient, if it is less than the minimum score the integrated camera cannot record an object. If the minimum score is too low, the algorithm will detect false positives. If the minimum score is too high, the algorithm will miss targets.

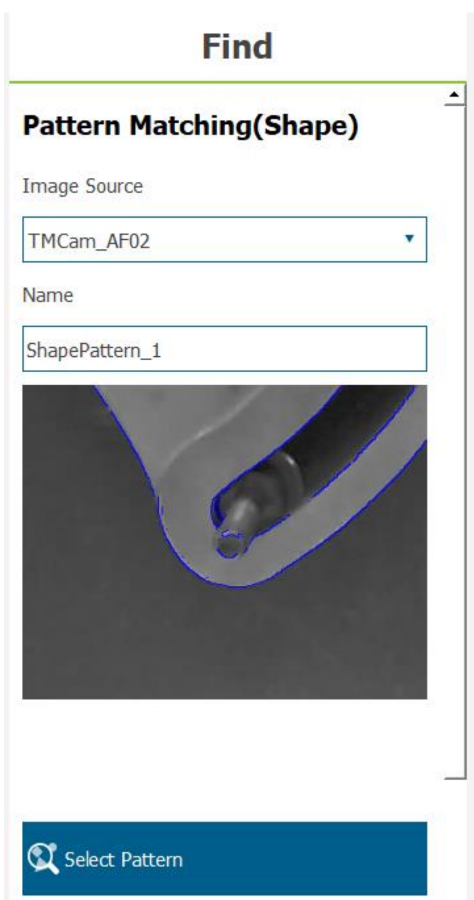

Figure 18

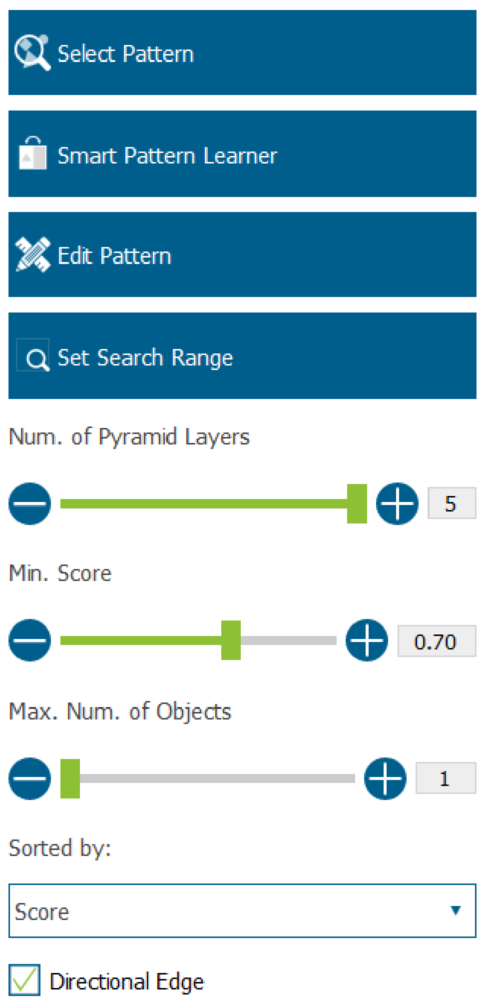

Figure 19

- Object Classification:

Following feature extraction, the detected objects may be classified into predefined categories or classes. This classification step enables the system to identify the type or nature of each object detected, which is crucial for subsequent decision-making processes.

- Localization and Tracking:

In addition to detecting and classifying objects, the Omron TM5-900 may perform localization and tracking tasks to determine the precise position and movement of objects within the robot's workspace. This information is vital for effective interaction and manipulation of objects by the robot.

- Decision-Making and Response:

Based on the results of object detection, classification, localization, and tracking (Figures 20, 21), the Omron TM5-900 can make informed decisions and execute appropriate actions. These actions may include picking and placing objects, navigating around obstacles, or interacting with the environment in predefined ways. As minimum score is almost higher than 0.90, so accuracy is high. After a number of tests, during recognition of the main part, precision is better in ±15° angle, if more than this, accuracy is preferable, by precision gets worth.

Figure 20: *Object detection*

Figure 21: *Object detection (rotated part)*

- Iterative Improvement:

Object detection with the Omron TM5-900 is often an iterative process, where the system continuously learns from its observations and refines its algorithms over time. This iterative improvement helps enhance the accuracy, efficiency, and robustness of the object detection system.

- Integration with Robot Workflow:

Finally, the object detection process is seamlessly integrated with the overall workflow of the Omron TM5-900 robotic platform. Object detection results inform and guide the robot's actions, enabling it to perform complex tasks autonomously and adaptively. Overall, the process of object detection with the Omron TM5-900 involves a combination of image acquisition, preprocessing, algorithmic analysis, decision-making, and integration with robotic workflows to enable intelligent interaction with the environment.

3.3 Use case of mechanical parts

In a typical assembly operation, the mechanical components involved often appear quite similar to each other. However, for the thesis, the cobot must distinguish one piece from another by identifying at least one technological feature that sets it apart. Consequently, distinct pieces have been selected that may appear similar at first glance but differ in at least one characteristic. For instance, as depicted in Figure 22, two steel gears have been chosen: they are the same shapes, same number of holes, same color and both are made of steel. The algorithm aims to recognize these two gears as two different objects.

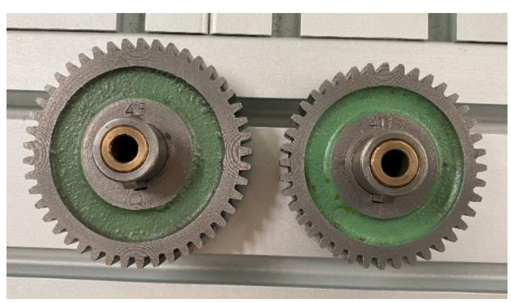

Figure 22: *Two similar gears*

Additionally, among the assembly pieces, there are two identical screws (Figure 23) that are needed to distinguish between the above-mentioned gears. The threads and gears have different shapes, numbers of holes, and colors, but both are made of steel. The main part of the assembly shown in Figure 24 is different from the other parts.

Figure 23: *Two identical screws*

Figure 24: *Main part of the assembly*

Below from Figure 25, is a fully assembled picture of the thesis' main goal.

Figure 25: *Assembled*

4. Complete sequence picking and placing

4.1 Complete sequence by vision

In the thesis, the assembly sequence is shown in Figure 26 step by step.

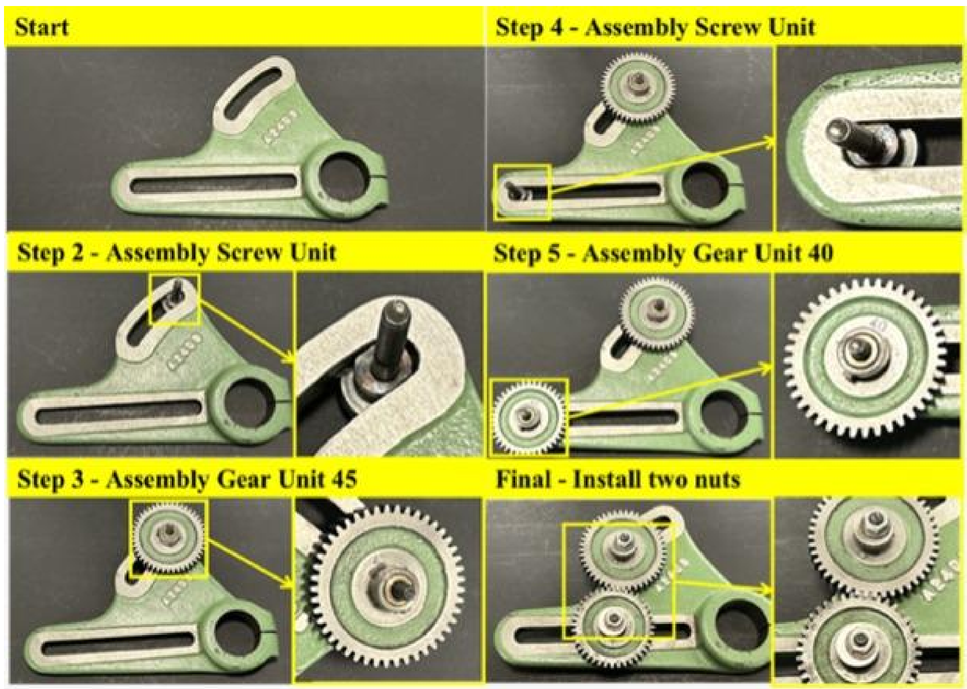

Figure 26: *Assembly sequence*

Before starting the cobot need to place parts as shown in Figures 22,23, and 24 horizontally in the working area. Step 1 is positioning the main part to the proper place, in the range of vision. The next step is recognizing one out of two of the screws in Figures 27 and 28, picking and placing it an in advanced programmed space.

Figure 27: *Servoing*

Figure 28: *Servoing finished*

The third step is object recognition, the cobot looks for the correct gear unit 45 from the initial point with the help of the vision node. After recognition, the cobot starts servoing to find the exact position of the barycenter to grasp it by paying attention to the moments of inertia to manipulate the piece correctly (Figures 29 and 30).

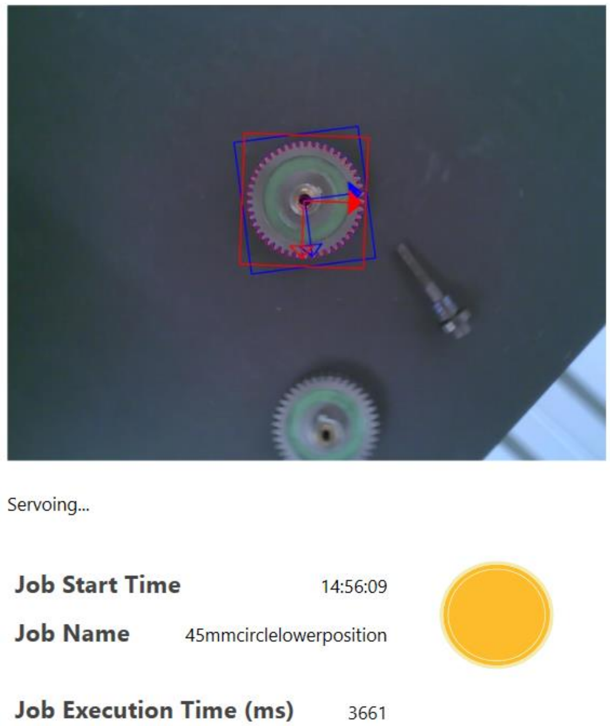

Figure 29: *Servoing*

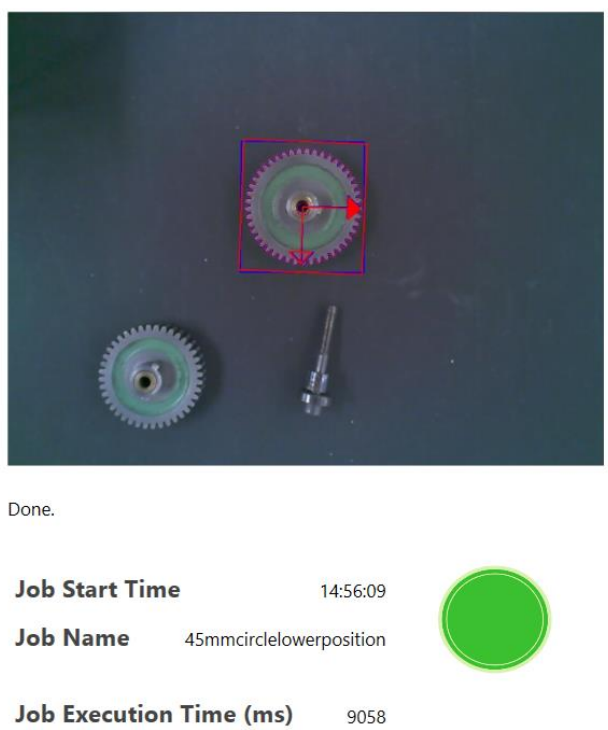

Figure 30: *Servoing finished*

During this time, human places screw that has brought by the cobot to the main part. Here need to be accurate to put the screw in the corner. If the screw is not at the position, the cobot fails with collision and stops. Then, continues grasping the piece and moves to the in advance mentioned position in the program. From that point, vision node object detection works (Figure 31).

Figure 31: *Frame detection*

The integrated camera helps to correctly recognize the proper corner of the main part and the screw. Here need to position the main part at the correct angle. As shown in Figure 31, this is the correct angle of positioning. As there are no FT 300-S force torque sensor [10], the cobot cannot sense an object during insertion, so it fails with collision. If there is a force torque sensor, it has a contact detection feature that allows smooth insertion. Additionally, there is an acceptable limit of ±15°. If it is not in the range of the limit, the cobot fails, because the center of the gear and the center of the screw cannot coincide.

Figure 32: *FT 300-S force torque sensor*

It starts inserting the gear into the screw by turning to some angle. The next steps continue in the same way (Figure 33).

Figure 33: *Recognizing frame to insert the second gear*

4.2 Complete code explanation

Below Figure 34, see the start of the flowchart of the pick&place program. Firstly, the cobot moves to the initial position and then checks the condition of Robotiq 2F-85 with a wide opening and closing of its fingers. In the red box, there is a node called the “Sub-flow” node. This node is needed to respect the sequence of assembly. The assembly sequence is a *screw - gear_45mm – screw – gear_40mm.* To respect the sequence, the code is programmed into two parts: main_flow and sub_flow.

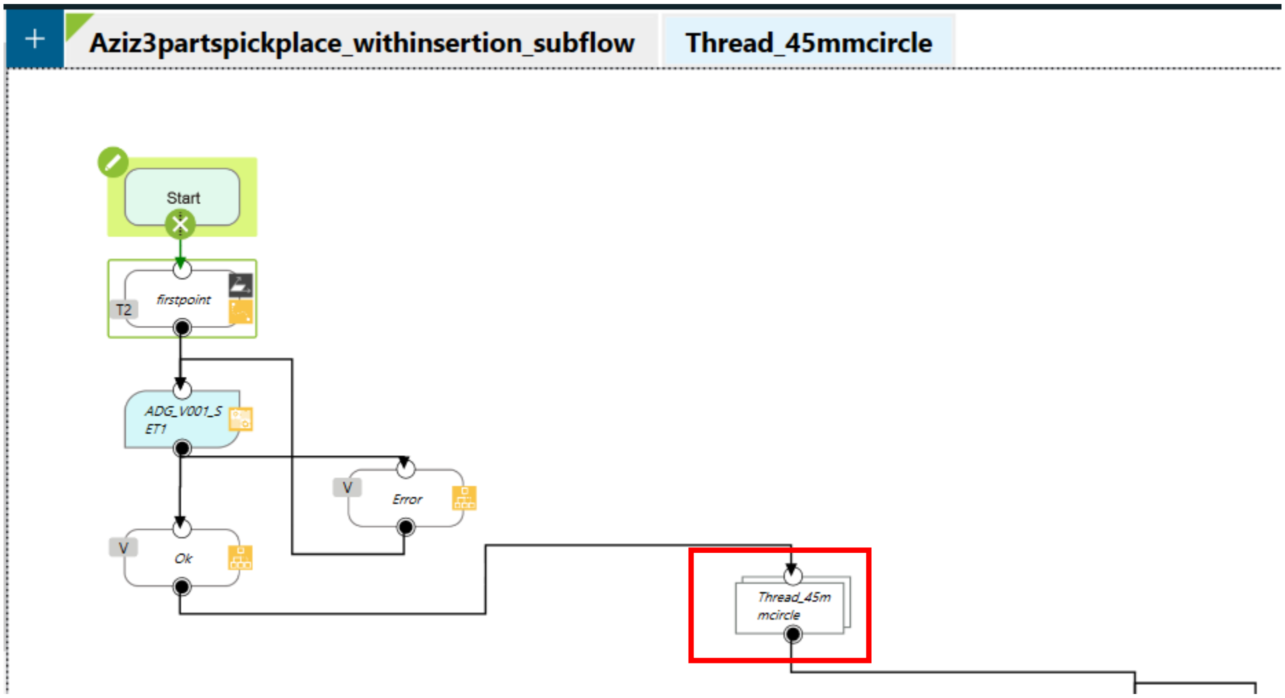

Figure 34: *Beginning of the code*

In sub_flow contains the first two sequences *screw – gear_45mm* and the main_flow contains *screw – gear_40mm*. Inside the red box, there is the sub-flow node which follows Figure 35. Figure 35 shows sub_flow which continues with the vision node "thread", this node has two outports "Pass" and "Fail". If the integrated camera cannot find, it fails and again tries object detection with by showing on display "NotFound". If it finds, it continues with moving to a specified point. Here this point is straight above the piece. In the point node (Figure 35) there are three motion settings (PTP, Line and WayPoint), from the vision node to the point node moving it is chosen PTP. Because of choosing PTP, it moves fast and from the shortest way.

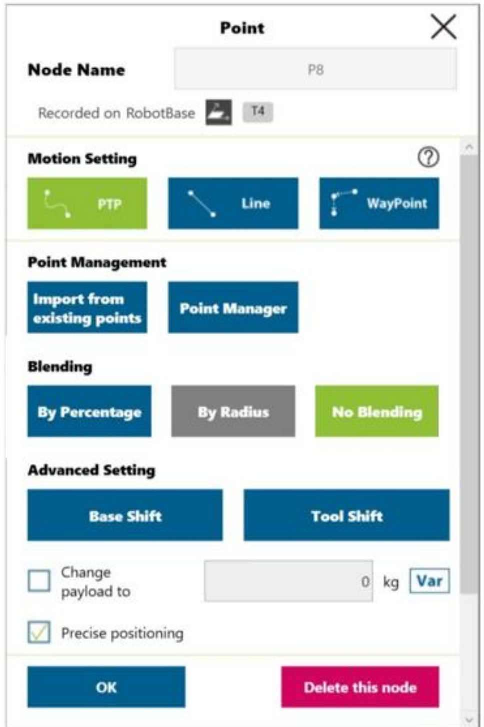

Figure 35: *Point node*

The next step is the "Move" node. It is also similar to the "Point" node, but in this case, it is preferable to choose the "Line" motion setting. After that reaching to determined position, the cobot grasps the screw and the flowchart continues from the "Detected" node (Figure 36). With grasping the screw, the cobot moves to a predefined place to release it. At the end of this step, there is a "Go to" node, that simplifies code by only defining its target node. Here its target node is the vision node for gear_45mm. Furthermore, the cobot checks the gripper and moves while doing other steps. In this sub-flow, there is one more task picking and inserting gear_45mm. This task is almost the same as the screw task. After placing the screw, it goes to the initial point to find and recognize gear. The main difference from the screw task is that in this task the cobot needs to insert a gear into the screw with some inclination (The right part of Figures 36 and 37).

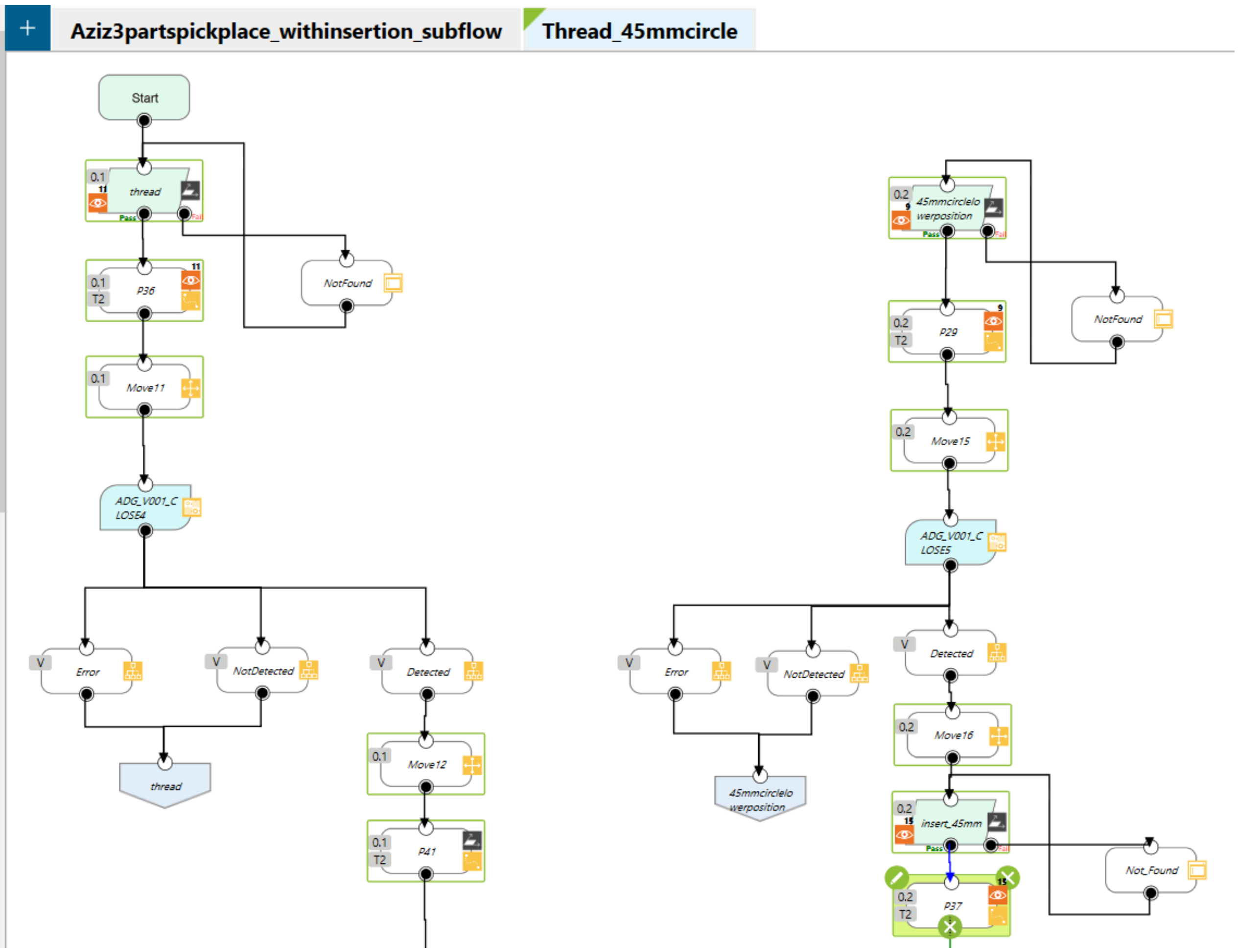
Aziz3partspickplace_withinsertion_subflow
Thread_45mmcircle
Start
thread
NotFound
P36
Move11
ADG_V001_C LOSE4
Error
NotDetected
Detected
thread
Move12
P41
45mmcirclelo werposition
P29
NotFound
Move15
ADG_V001_C LOSE5
Error
NotDetected
Detected
Move16
45mmcirclelo werposition
insert_45mm
Not_Found
P37

Figure 36

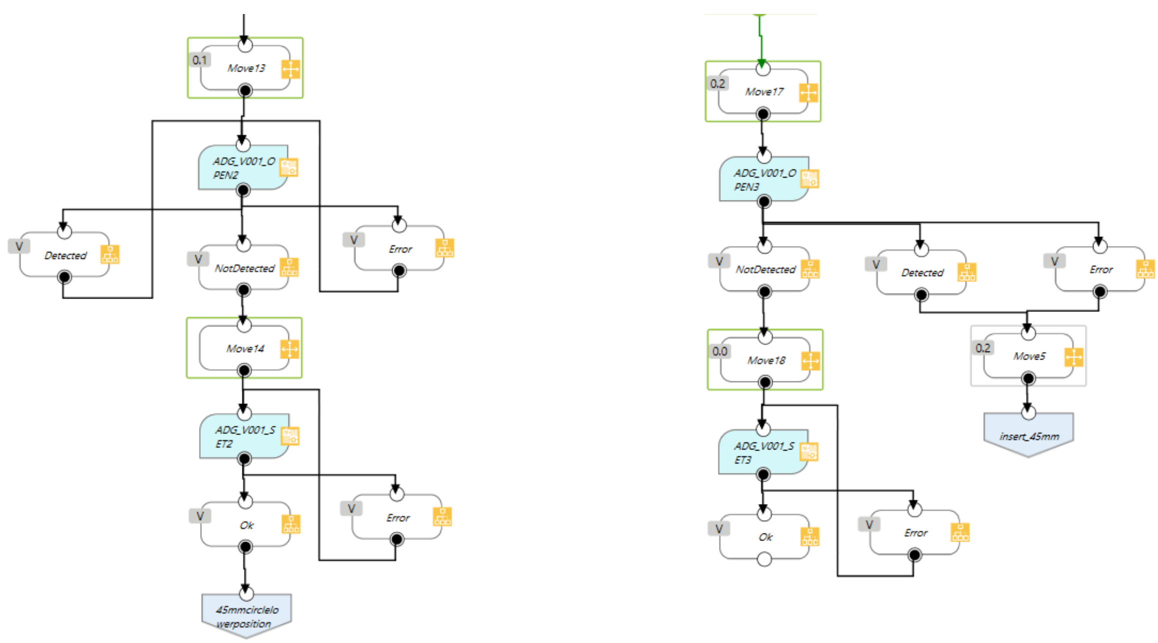
Move13
ADG_V001_O PEN2
Detected
NotDetected
Error
Move14
ADG_V001_S ET2
Ok
Error
45mmcirclelo werposition
Move17
ADG_V001_O PEN3
NotDetected
Detected
Error
Move18
Move5
insert_45mm
ADG_V001_S ET3
Ok
Error

Figure 37: continue Figure 36

After assembling one screw and one gear, need to proceed other steps which are the second screw and the smaller gear 40mm (Figures 38 and 39).

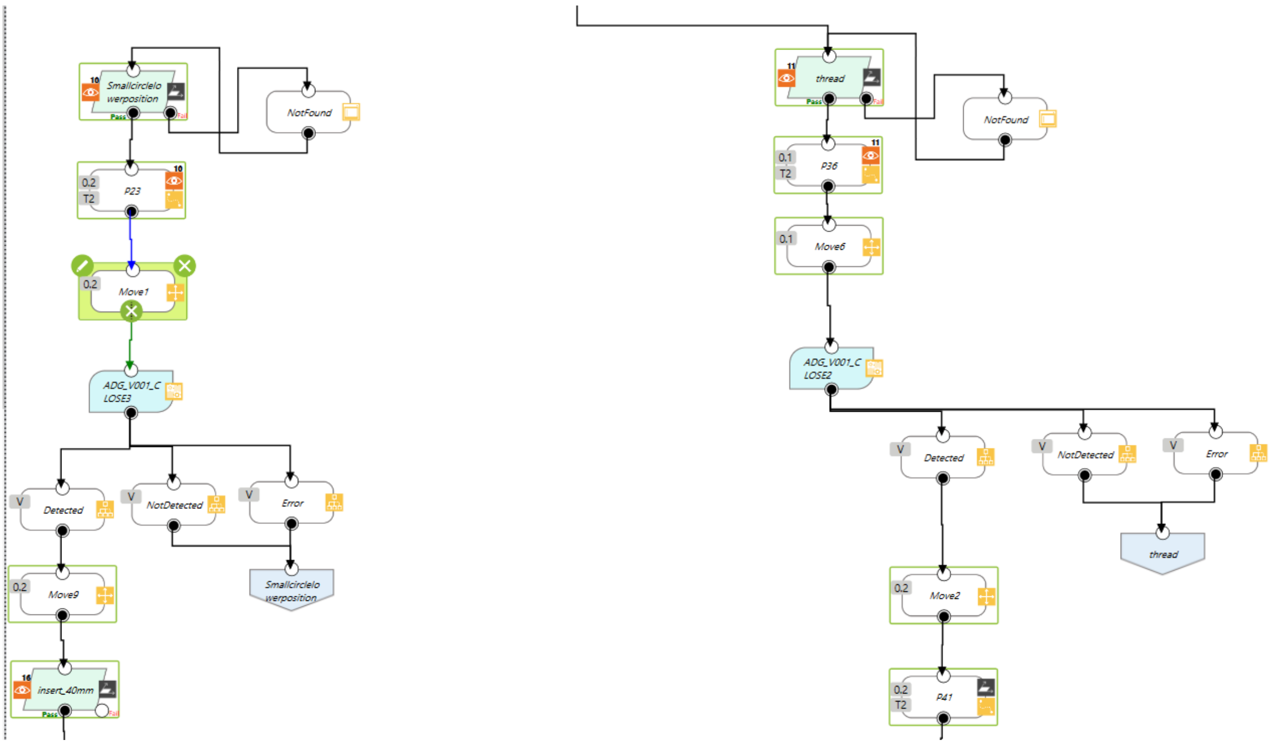

Figure 38: continue Figure 37

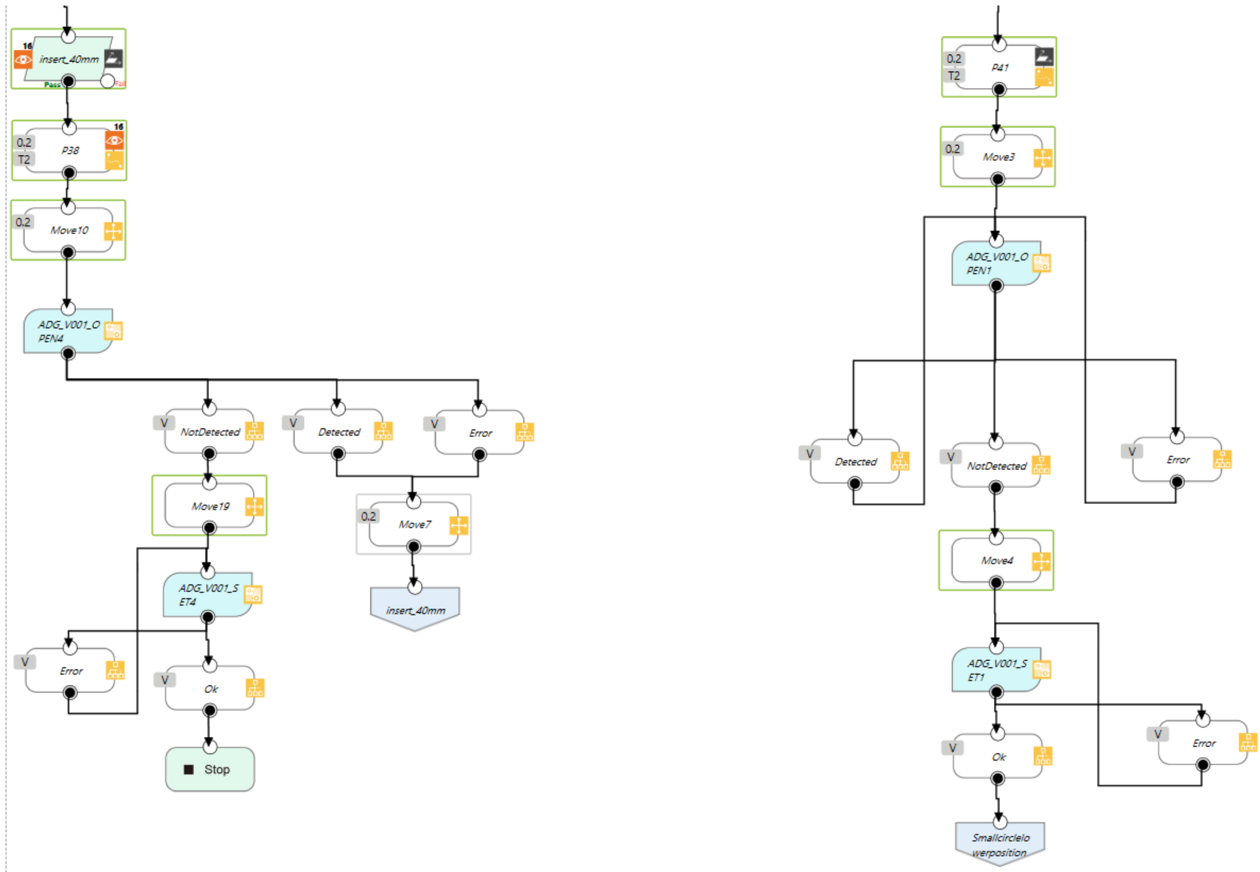

Figure 39: *continue of Figure 38*

5. Accuracy and working of the system

5.1 Possible mistakes by humans

Throughout the testing phase of the collaborative robot (cobot) in an unstructured environment, a series of tests were conducted, revealing various types of mistakes. Approximately 20 tests were executed to define possible mistakes by human (Figure 40), each highlighting common errors such as misalignments, object misplacements and assembling sequences. The cobot underwent rigorous testing in unstructured environments, consisting of 25 tests. Analysis of these tests revealed recurrent mistakes by cobot (Figure 41), including collision incidents, sensor inaccuracies, and failures. A total of 30 tests were performed to assess accuracy and precision. These tests provided insights into the diversity of errors observed, with recurring issues such as object misplacements, sensor malfunctions, and occasional system errors. After resolving all the mistakes that came from tests, again 20 tests conducted. In these tests, by following instructions the tests completed without errors.

№	Possible mistakes by human	Reason	Mistake avoidance
1	Placing parts: not placing parts in a range of cobot view	Human is not familiar with instruction. The cobot programmed only to search from one position.	This mistake can be avoided with reprogramming: the need to give several positions to search for an object. It takes more time to look for parts
2	Placing parts: some parts are in the wrong position (for example thread: don’t put them vertically)	Human is not familiar with instruction. The cobot will stop immediately as there is a limit for grabbing parts.	Follow instructions to put parts correctly.
3	Placing parts: some parts are in the wrong position (for example gears: don’t put it vertically)	Human is not familiar with instruction. The cobot will stop immediately as there is a limit for grabbing parts.	Follow instructions to put parts correctly.
4	Assemble screw: Human does not assemble screws with the main part	Human is not familiar with assembling sequences.	Follow assembling sequence
5	Inserting gear: not placing the main part in the range of the cobot view	Humam is not familiar with instruction. The cobot programmed only to search from one position.	This mistake can be avoided with reprogramming: need to give several positions to search for an object. It takes more time to look for parts

6	Insert gear: not placing the main part in a right angle (±15° acceptable)	Human is not familiar with instruction.	To successfully insert gears, respect the instruction main part positioning with only ±15°. The second way of reprogramming with adding to the Cobot FT 300-S Force torque sensor [10].
7	Insert gear: not placing the thread in the right position (in the corner)	Human is not familiar with instruction.	Follow instruction

Figure 40: *Possible mistakes by human*

5.2 Possible mistakes by Cobot

№	Possible mistakes by Cobot	Reason	Mistake avoidance
1	Insert gear: recognizing the main part with the wrong position of the screw (moved a bit or more)	The integrated camera cannot recognize the base of the screw. In this program, the cobot recognizes the frame of the main part.	With using FT 300-S force torque sensor [10], the cobot can contact with an object. During contact, depending on the value of force it corrects its angle and position.
2	Insert gear: not placing the main part at a right angle (±15° acceptable)	As there is no FT 300-S force torque sensor, the center of the gear does not coincide with the center of the screw	With using FT 300-S force torque sensor [10], the cobot can contact with an object. During contact, depending on the value of force it can find an axial axis to insert.
3	If an error occurs wherever during the application with collision, it needs to be started from the beginning	The cobot's safety regulation makes the robot stop immediately in a collision.	

Figure 41: *Possible mistakes by Cobot*

Conclusions

In conclusion, this thesis aimed to develop a computer vision algorithm tailored for implementation in a collaborative robot (cobot), to enable accurate identification and manipulation of various mechanical components within unstructured environments. The successful execution of robotic assembly tasks in such environments hinges on the cobot's ability to select and handle components in a precise sequence, even when their positions are unknown or variable. By utilizing integrated cameras and sophisticated algorithms, the thesis achieved significant progress in streamlining the assembly process. The computer vision algorithm demonstrated proficiency in distinguishing between similar mechanical components based on key technological features. Moreover, the integration of TMFlow software with Omron cobot facilitated precise barycenter localization and manipulation, ensuring efficient assembly operations. Overall, this research contributes to the advancement of automation in industrial settings by enhancing the capabilities of collaborative robots in unstructured environments. The developed computer vision algorithm holds promise for improving efficiency and adaptability in various assembly tasks, ultimately enhancing productivity and reducing reliance on pre-established programming.

Sitography

[1] https://www.ibm.com/topics/object-detection

[2] https://machinelearningmastery.com/object-recognition-with-deep-learning/

[3] https://godelius.com/en/the-future-of-robotics-in-unstructured-environments/

[4] https://www.static.tu.berlin/fileadmin/www/10002220/Publications/Katz-08b.pdf

[5] https://robotics.omron.com/collaborativerobots/tutorial-videos/

[6] https://www.ia.omron.com/products/family/3739/specification.html

[7] https://automation.omron.com/en/us/products/family/Omron%20TM%20Robots

[8] https://en.wikipedia.org/wiki/International_Federation_of_Robotics

[9] https://builtin.com/robotics/cobot

[10] https://robotiq.com/products/ft-300-force-torque-sensor

[11] https://robotiq.com/products/2f85-140-adaptive-robot-gripper