UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Design, development and evaluation of a 3D Vision system for the monitoring of automatic assembly and mounting operations.

Master's Thesis

Master's Degree in Industrial Engineering

AUTHOR: Dañobeitia Capetillo, Kerman

Tutor: Sánchez Salmerón, Antonio José

Cotutor: Valera Fernández, Ángel

ACADEMIC YEAR: 2024/2025

DESIGN, DEVELOPMENT, AND EVALUATION OF A FLEXIBLE MANUFACTURING CELL USING VISUAL GUIDANCE FOR COLLABORATIVE ROBOTS FOR AUTOMATIC ASSEMBLY AND MOUNTING OPERATIONS

MASTER THESIS
Academic Year: 2024/2025

MASTER OF SCIENCE IN ELECTROMECHANICAL ENGINEERING
Universiteit Gent

MASTER UNIVERSITARIO INGENIERÍA INDUSTRIAL
Universidad Politécnica de Valencia

Author: Kerman Dañobeitia Capetillo

Univeristeit Gent Promotor:
Dr. Frederik Ostyn

Universidad Politécnica de Valencia Promotor:
Antonio José Sánchez Salmerón & Ángel Valera Fernández

Acknowledgments

As the author of this master's dissertation, I would like to take this opportunity to express my sincere gratitude to everyone who has contributed to making this work possible.

First, I extend my thanks to my companion on the project *Unai Amenabar*. He was responsible of developing the robotic part of the project and through all the year we worked alongside each other, helping each other out when it was needed and taking the project's development as far as possible inside our capabilities.

On the same way, I also want to share my gratitude towards all the tutors of the project that gave me guidance and support through the journey. Their expertise, constructive feedback, and guidance were essential to the success of this work. Therefore, I am deeply grateful to *Antonio José Sánchez Salmerón*, who was my main tutor and the expert on the Artificial Vision theme and to *Ángel Valera Fernández,* my second tutor expert on robotics. They both helped out considerably in the project and without them the project would not reach its actual extend.

Working on this dissertation has been an incredible journey for me, filled with both challenges and opportunities. Throughout this process, I learned the significance of persistence, patience, and commitment in achieving my goals. Looking back on this experience, I am filled with a sense of accomplishment. I hope this work will contribute meaningfully to the ongoing discussions and research in this field.

Lastly, I would like to express my sincere appreciation to all those who have supported and contributed to this work in a different way. A special thank goes to my family and friends, whose encouragement and support kept me focused and motivated.

Thank you all.

Admission to Loan

Kerman Dañobeitia, June 27

Abstract

In the era of Industry 5.0, optimizing production processes through human-machine collaboration is fundamental. This project focuses on applying artificial vision and deep learning in robotics for automating complex assemblies. The study's purpose was to design and evaluate a manufacturing cell for autonomous 3D assembly of randomly placed elements, using vision-guided collaborative robots, aiming for an adaptable, optimal industrial procedure.

Methodologically, the system employs two UR3e robots and an Intel RealSense D455f 3D camera in an environment with randomly distributed components. An instance segmentation model (YOLO) was used for object detection and characterization. The procedure included developing algorithms for 3D localization of optimal gripping points, determining orientation, and dynamic generation of the picking order. Communication between the vision system and robot controllers was managed via a robust command-based protocol over Wi-Fi.

The results demonstrate high performance. The instance segmentation model achieved 98.98% accuracy in object detection. Meanwhile, 3D localization precisely extracted size, position, and rotation angle; a specific module for hole identification achieved 99.58% accuracy. Reliable communication and real-time performance optimized to 30 FPS were crucial. However, camera inaccuracy (±3.3mm) exceeded the robot's tool tolerance (±2.5mm), indicating a significant limitation.

In conclusion, the feasibility of a flexible assembly cell for random elements was demonstrated, fulfilling the primary objective. The project lays solid foundations for automation, though areas for improvement are identified: the need for higher camera precision for full robotic integration, the potential of more comprehensive 3D vision, and the implementation of error detection systems. This work offers a valuable proof of concept for optimizing industrial productivity and complexity.

Keywords: Industry 5.0, Collaborative Robotics, Automatized systems, 3D Artificial Vision, 3D Assembly, Object identification and classification, Neural Networks, Python, Intel RealSense

Resumen

En la era de la Industria 5.0, optimizar los procesos de producción mediante la colaboración hombre-máquina es fundamental. Por ello, este proyecto se centra en la aplicación de la visión artificial y el aprendizaje profundo en robótica para la automatización de ensamblajes complejos. El propósito del estudio fue diseñar y evaluar una célula de fabricación para el ensamblaje 3D autónomo de elementos colocados aleatoriamente, utilizando robots colaborativos guiados por visión en tiempo real, buscando un procedimiento adaptable y óptimo para la industria.

Metodológicamente, el sistema emplea dos robots UR3e y una cámara 3D Intel RealSense D455f en un entorno con componentes distribuidos al azar. Se utilizó un modelo de segmentación de instancias (YOLO) para la detección y caracterización de objetos. El procedimiento incluyó el desarrollo de algoritmos para la localización 3D de puntos de agarre óptimos, la determinación de la orientación y la generación dinámica del orden de recogida. La comunicación entre el sistema de visión y los controladores del robot se gestionó mediante un protocolo robusto basado en comandos vía Wifi.

Los resultados demuestran un alto rendimiento. El modelo de segmentación de instancias alcanzó una precisión del 98.98% en la detección de objetos. Mientras tanto, la localización 3D extrajo con precisión el tamaño, la posición y el ángulo de rotación; un módulo específico para la identificación de orificios logró una precisión del 99.58%. La comunicación fiable y un rendimiento en tiempo real optimizado a 30 FPS fueron cruciales. Sin embargo, la imprecisión de la cámara (±3.3mm) superó la tolerancia de la herramienta del robot (±2.5mm), lo que indica una limitación significativa.

En conclusión, se demostró la viabilidad de una célula de ensamblaje flexible para elementos aleatorios, cumpliendo el objetivo principal. El proyecto sienta bases sólidas para la automatización, aunque se identifican áreas de mejora: la necesidad de mayor precisión de la cámara para una integración robótica completa, el potencial de una visión 3D más exhaustiva y la implementación de sistemas de detección de errores. Este trabajo ofrece una valiosa prueba de concepto para optimizar la productividad y la complejidad industrial.

Palabras claves: Industria 5.0, Robótica Colaborativa, Sistemas Automatizados, Visión Artificial 3D, Ensamblaje 3D, Identificación y Clasificación de Objetos, Redes Neuronales, Python, Intel RealSense

Index

Image index

Table Index

Symbols and abbreviations

UPV	Universidad Politécnica de Valencia (Polytechnical University of Valencia)
SME	Small and Medium Enterprises
DISA	Departamento de Ingeniería de Sistemas y Automática (Department of Systems and Automation Engineering)
SDG	Sustainable Development Goals
ToF	Time-of-Flight
IR	Infra-Red
RGB	Red, Green and Blue
RMS	Root Mean Square
IMU	Inertial Measurement Unit
FOV	Field of View
FPS	Frames Per Second
ROI	Region Of Interest [Computer vision]
HD	High Definition
VGA	Video Graphics Array
ASIC	Application-Specific Integrated Circuit
TCP	Tool Centre Point
JSON	JavaScript Object Notation
NN	Neural Network
AI	Artificial Intelligence
YOLO	You Only Look Once
SSD	Single Shot Detector
DETR	DEtection TRansformer
COCO	Common Objects in Context
mAP	Mean Average Precision
ROI	Return Of Investment [Economic]

1. Introduction

1.1 Problem quoting

One of the most important areas of industrial companies is production, whose activity has a direct impact on the quality and price tag of the final product. Also, these two aspects of the generated output, contribute mainly to the satisfaction of the customer, so in order for a company to be competitive in the market, the productivity must be optimized at all times. Although this project will not be carried out for a specific company or process, the developed process/technology is aimed to be applicable to any manual assembly.

Usually, one of the most important costs of the production line is the time the worker needs to assembly the product. The duration of this assembly starts when the minimal number of pieces to do the assembly is available and ends when the desired final element is mounted. In most cases, the production company cannot directly control the availability of the required material, so the duration of the operator's assembly is the critical point of production. All companies involving only human personnel in the assembly follow a common process similar to the one shown on Image 1(with little changes from company to company).

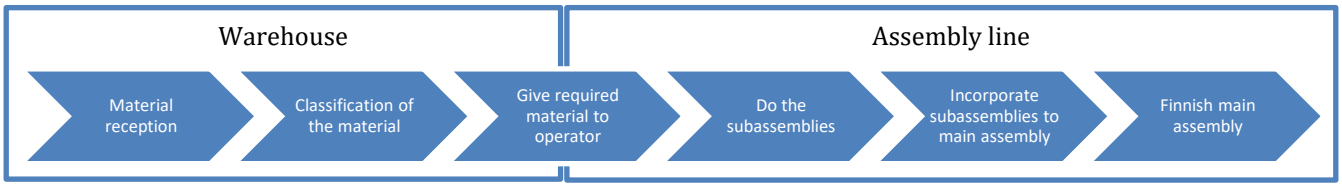


Image 1: Assembly process

Specially, when complex assemblies are performed, more operators and special equipment are required. This produces other technicians not being able to carry their job at full capacity, reducing the overall productivity of the company. Or else way, having to buy more stock of that required special equipment, which in most cases costs a lot of money.

1.1.1 Industry 5.0

The tendency of the industry in the last years has been towards automating every aspect of a company and replacing workers directly with robots or other mechatronic systems that can do the same work an operator can do. This is known as Industry 4.0, and yes, it achieves more efficient productivity and products with higher quality, but a lot of basic level workers lose their job because of them being replaced. In this context, the concept of Industry 5.0 arose during the Covid pandemic. This whole evolution process of the industry can be seen on the Image 2. According to the European Commission *[1]*:

"*The Industry 5.0 approach provides a vison of industry that aims beyond efficiency and productivity as the sole goals and reinforces the role and the contribution of industry to society.*

It places the wellbeing of the worker at the centre of the production process and uses new technologies to provide prosperity beyond jobs and growth while respecting the production limits of the planet.

It complements the existing "Industry 4.0" approach by specifically putting research and innovation at the service of the transition to a sustainable, human-centric and resilient European industry."

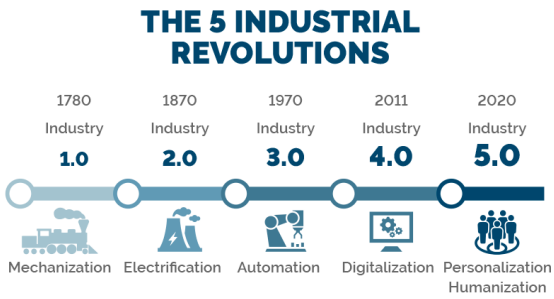


Image 2: A new phase of the industrial revolution *[2]*

1.1.2 Situation in the region

Considering all these, the team has figured out that there is an interesting improvement opportunity on the market, especially in the Spanish industry, where most of the jobs performed nowadays have huge workloads for the operators or have very low levels of technology application. This is a consequence of the scarce belief of the *Small and Medium Enterprises* (*SME*) on the necessity to invest in modern technologies. As the news webpage *CincoDías* states: "*34% of SMEs do not plan to invest in technology*" *[3]*, which is going against the global trend to invest a higher percentage of the total incomes towards new technology implementation, shown by the research made by Deloitte on the Image 3.

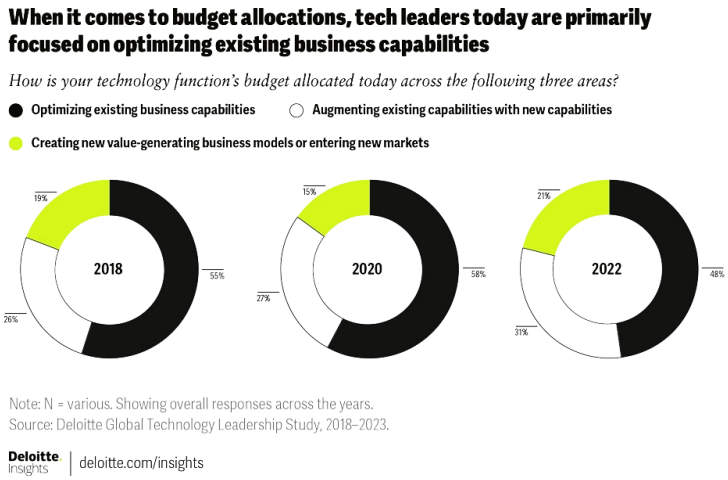


Image 3: How tech leaders allocate their tech budgets *[5]*

Also, knowing that the industrial sector of Spain in composed in 99,4% by SMEs *[4]*, this means that one third of the companies dedicated to the industrial sector do not consider investing in technology. Although this will not have a direct impact in the short-term, on the long run this will suppose a significant loss of opportunity for keeping the national industry up to date and efficient in the global market, where more and more countries are putting their focus on this aspect and taking advantage of this market opportunity to make their industries globally advanced and competitive. Therefore, the necessity in Spain to raise awareness and find solutions or alternatives for those companies that are not willing to invest in technology must be primordial. And the Industry 5.0 can be the key to achieve this

As the concept of the new industrial revolution focuses on the operators and workers instead of just caring about productivity in general, the new developments tend to generate cheaper products and services. There is no necessity to create an entire automated manufacturing line if there is already a process based on operators that works correctly and just needs little tweaks. This mindset change leads to cheaper productivity optimization and satisfaction of the workers, who are more trusted; two factors that are usually the most concerning ones when the decision of an investment needs to be made.

1.1.3 Application

Considering the current Spanish industry's situation and its future tendencies in the global market and the opportunity that is arising with the new industrial revolution, it has been decided that the best decision for this Master Thesis will be developing a new collaborative technology that can help companies implement the industry 5.0 on their own enterprises, making them more competitive and technologically advanced.

Exactly, the development of a "cheap" aid technology that makes the operator's job more comfortable while building on the productivity raise is the aim of this project. This technology should be applicable to a range as broad as possible, meaning that it should be able to help with nearly any kind of mechanical assembly that each enterprise may have, and must not mean a significant investment to any SME whereas they still obtain a big benefit from its implementation.

As this project has its own time and resource limitations it will just develop a first prototype. Therefore, a specific product has been selected as an assembly base: a wooden chair. This product is a simple start that can also show all the different key aspects that the project will need to develop for being applicable to other assemblies and sectors. Afterwards, this same functioning principle should be applied to assemble any other wooden furniture with similar scale. Next step will be to translate the technology to other industrial sector such as the metallurgical (which require far more accurate precision). The reach of the project will depend on the available time and resources obtained during its development.

1.2 Innovations

At the moment that the project has started, there are already some applications that have achieved automatic furniture assemblies using robots, such as:

- Huayan Robotics, 2023 [6]. Uses one unique robot without an artificial vision system to pick and place the wooden elements in an assembly machine. This machine has positioners installed so that all elements are always placed in the same position. To insert the dowels into the elements and execute the assembly operations, the machine relies on cylinders. All in all, although it may be a less costly machine due to the unique robot, it requires significant space to operate and a considerable number of auxiliary elements. Furthermore, it does not fulfil the entire assembly of the chair, but only one side of it. The main advantage of this application is the duration of the assembly, which barely takes 30 seconds, however, if the holes of the dowels are not done exactly in the required position, the machine will fail. Considering that the wooden elements usually do not meet achieve very high precisions (at around ∓ 2[mm]), the failure rate of the machine can turn out to be high.
- Nanyang Technological University, 2018 [7]. The first fully assembled chair using two collaborative robots with 3D cameras incorporated. According to newspaper ASSEMBLY [8], "a pair of six-axis

Denso robots assembled a Stefan chair in 8 minutes and 55 seconds. Prior to the assembly, the robots took 11 minutes and 21 seconds to independently plan the motion pathways and 3 seconds to locate the parts." The articles and videos on the Internet do not show the entire assembly process, but they do show that the assembly is done without auxiliary positioning systems, with one robot holding the elements while the other assembles the dowels or required element.

- COEX, 2021[9]. Uses three robots (it is not specified if they are collaborative or industrial) and a rotatory table with cylinders for movement in the *Z* direction of the assembly. It does not include a artificial vision system, so it requires all elements to be in fixed position for picking. According to the video, the process takes about 30 minutes to complete.

If we compare the three most popular applications similar to the one suggested in this project, it can be observed that either they take a lot of time to do the assembly, or they require auxiliary systems and huge working space. Both problems mean a lot of money in the eyes of the enterprises, which may have led this application to fail in the industry.

Nevertheless, apart from the example solutions mention above, the team has not found other companies or research teams that have achieved a fully automatic assembly of a wooden chair with and without human intervention. This project will aim to improve the efficiency and duration of the previous solutions. In order to achieve this, a new set of ideas and simplifications are tested:

- Only one auxiliary system. Responsible for the positioning of the bolts and dowels in an upwards position so that the robot can pick them from the head. Even though it means manufacturing additional elements, it can significantly decrease the total duration of the assembly process. Other systems are considered as not necessary if more than one robot is available as in these cases the robots themselves can act like references for the assembly operations.
- Only one artificial vision camera. In the solution developed by Nanyang Technological University, 2018 [7], each robot has a 3D camera mounted in the tool frame. These cameras are used to locate the holes of the dowels in the assembly surface and to identify the picking elements and define their orientation. However, the team considers that an algorithm can be developed so that the exact location of the assembly holes is not necessary, but only an approximation of it. Then, by using the force and position sensors of the robots, they will be able to identify when a hole is found. Therefore, the only function of the camera would be to identify the objects in the working area and define the position. And for so, there is no need to use two 3D cameras. Using one 3D camera or designing a stereo vision system with two 2D cameras would be enough in that case.
- Elements of interest in different planes. In both applications where the artificial vision system is used [7, 9], all the objects that make the assembly are placed in the same plane, scattered in the working area of the robots. Such disposition means that the cell total space is considerable just to place all required elements in the working area. In this case, taking advantage of the 3D camera, different elements can be placed on top of each other and then, if the system requires a certain element that is not reachable due to other elements being on top, the robot will be able to discard those elements to some prefixed positions, outside the range of the artificial vision system.

With these three main concepts, the team thinks that the duration and cost of the application can be decreased to a point where it can be applicable to industry, specially Small and Medium Enterprises.

1.3 Background

In order to develop the project, the department of systems and automation has provided de project team (two Master students) with a set of elements from the same department, as well as some material from the Machine Vision department and access to Robotics Laboratory in the DISA department building (UPV Campus de Vera building 5E) and all the material in the room (computers, wooden elements for puzzles, etc.).

As the department is currently focused on robotic applications on Spanish industry, the main lent elements are the four robotic arms that are currently available in the Robotics Lab, an in-depth description and analysis on these robots is done in the *State-of-the-Art* section:

- ABB IRB 140
- ABB IRB1100
- UR3e

Obviously, each robotic arm comes with its controller and certification to use the programming software (*RobotStudio* and *PolyScope* + *URSim*), installed on the computers of the lab. However, the access to a virtual machine with the required software has also been given so that the team members can work when the lab is not available due to lectures.

Also, as the tool of the robotic arm needs to be design specifically for the intended application, a 3D CAD software has been made available in one of the computers in the laboratory: *SolidWorks*.

Another aspect of the project is the machine vision system. For this, the machine vision department was contacted for available material, but finally opted to do market research by the project team. Choosing to buy the desired material so that no research is disturbed in other university areas. The analysis made to decide which camera and system to use is also done in the *State-of-the-Art* section.

1.4 Objectives

The main objective of this combined problem is to complete a 3D assembly of some randomly placed elements on a predefined working area, by means of using two collaborative *UR3e* robots that will do the assembly and a 3D *Intel RealSense* camera that will analyse and inspect the elements and workspace in real time to ensure an optimal procedure, so that later it could be adapted to a real life complex assembly process that any enterprise could have, reducing (an approximate of 50%) their production time, cost and complexity.

To achieve this main objective, some main tasks need to be defined for the computer vision area:

- Select the camera placing method and optimal point. As well as its working environment
- Tune it to the surrounding conditions to suppress all the noise caused by external factors and identify the elements of the assembly as accurately as possible.
- Being able to identify the different objects of a predefined assembly in 2D (from a backup list)
- Implement this system into a 3D model, being able to better distinguish the element's shape, position and orientation.

- Calculate the optimal point to hold each element and send this exact location to the robot along with the necessary orientation to do so.
- Create an algorithm that defines the picking order of the elements, so that the assembly follows a stablished pattern.
- Improve this algorithm so it completes this task with randomly placed initial elements, identifying the objects in an irregular environment and deciding the optimal order (not preestablished)
- Examine the results of the assembly to determine its quality and identify the defects made in this process with an accuracy of 80%

1.5 Planification

To achieve all the stablished objectives, it is essential to stablish and follow a proper planification that ensures the correct completion of the project within the stablished deadline. For this reason, all the main tasks and milestones have been identified and analysed following the PMI methodology and obtaining from it an initial Gantt [Image 4]. All the followed procedure to arrive till this planification are shown on the Annex A.

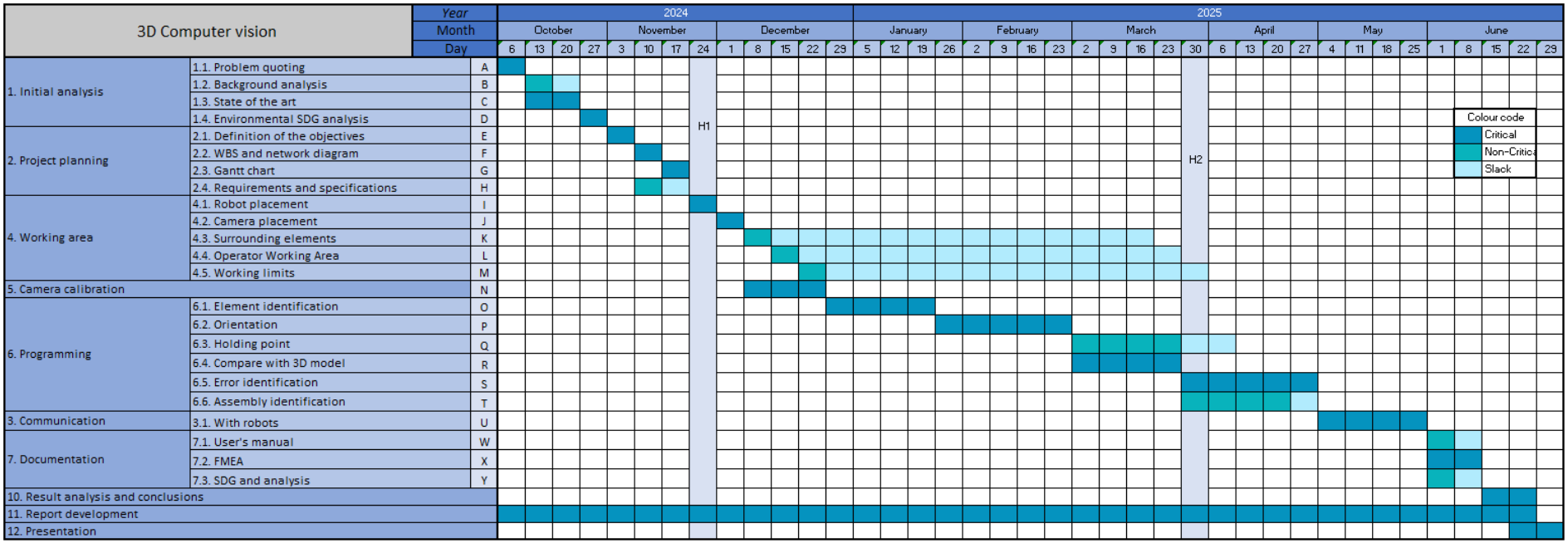


Image 4: Initial Gannt

1.6 Specifications

Once it has been decided which are the exact materials and technologies that are going to be used, the team has decided to further detail the functionalities and specifications to be developed.

As this is a first prototype of the product, the range of application needs to be limited for time and knowledge reasons. Therefore, the project will be applied to the assembly of a wooden chair, exactly the ODDVAR stool from IKEA *[10]*.

1.6.1 Initial conditions

First, the initial conditions are set, graphically explained in Image 5:

- Wooden elements are put in random places throughout the working table.
- Wooden dowels, screws and corner brackets are provided to the operator, instead of the robot.
- Two collaborative robots in standing position, each on one side of the middle of the working table
- 3D camera on top of the centre of the working table, out of the working range of the robots.
- Operator in front of the working table

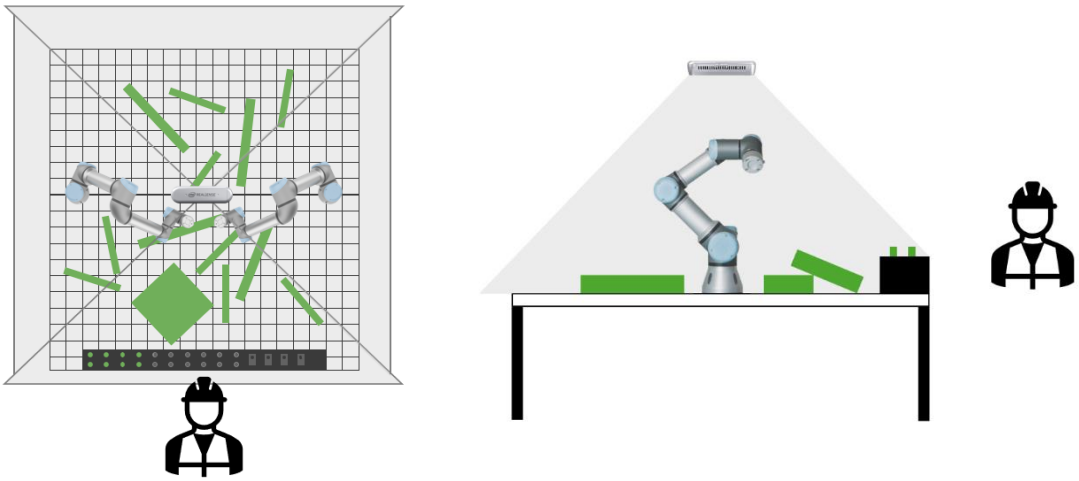


Image 5: Initial conditions of the working space

1.6.2 Functionalities: Robotics

Considering the starting situation, the robots are required to fulfil the next functionalities to ensure a proper application and implementation of the industry 5.0 principles:

- Collision detection through torque/force sensing. Thanks to the built-in force and torque sensor system, the robot must be able to stop when an unexpected resistance or force is detected. The robot should interpret it as a potential collision and immediately halt any active operation.
- Mechanical assembly based on torque/force sensing. When doing operations of assembly of wooden elements, they are usually attached or guided with wooden dowels that need to be put into position applying a certain force in order not to break the wood nor entering the dowel enough.

- Capability of identifying operator's position and direct element towards it. When the robot is required to do complex tasks as screw or bolt tightening, the motion needs to very precise and the gripper must also be designed specifically to deal with it. In order to keep the design simple and lowering the precision requirement, the robots must direct the elements towards the worker so that it can do the task more easily.
- Real-time communication with machine vision system. The system must ensure the correct assembly at each step of the assembly process, which changes the working area's state after each operation.
- Trajectory modification based on information from machine vision system. Also connected to the previous point, as the working state changes at each step, the predefined trajectories may adjust if something changes.
- Synchronization between both robots to do complex assemblies. Some operations require the use of two hands or robotic arms, so both robots must be coordinated precisely to perform those desired movements.

1.6.3 Functionalities: Machine vision system

Considering the starting situation, the machine vision system is required to fulfil the next functionalities to ensure a proper application and implementation of the industry 5.0 principles:

- Identification of each element based on comparisons with 3D CAD model. Although the machine vision system is able to identify each element through RGB and depth points, this is not very precise. To improve the performance of element picking, the information from the camera must be compared to a 3D model to automatically calculate the gravity centre of each element.
- Step by step monitoring of the assembly process. The system must ensure the correct assembly at each step of the assembly process, which changes the working area's state after each operation.
- Detection of anomalies or errors in the assembly elements and/or process. The system must be able to identify any kind of manufacturing error in the elements to avoid any undesired element crash.
- Detection of the operator's position at all times. To avoid any collision with the operator, although the robot already has collision detection, it is always safer if this is directly avoided through position monitoring.
- Real time communication with robotic system. The robot must know all the target points to pick the elements and do the desired operations. These points should be proportioned at each assembly step and provide the necessary information if any adjustment should be made to the trajectories.

1.7 Sustainable development goals

The sustainable development goals, or its acronym SDG, are goals adopted by all United Nations Member States in 2015 to complete with the sustainable development agenda of 2030. These goals intend to provide a shared blueprint for peace and prosperity for people and the planet, now and into the future. At its heart are the 17 Sustainable Development Goals (SDGs), which are an urgent call for action by all countries (developed and developing) in a global partnership. They recognize that ending poverty and other deprivations must go together with strategies that improve health and education, reduce inequality, and spur economic growth. All while tackling climate change and working to preserve our oceans and forests. *[11]*

These 17 goals are not only some bureaucratic targets that the UN has stablished for its own projects and the member counties' governments. They are also some especially important aspects to consider by everyone and to be applied in every project if we want to achieve a better world, because it is just possible to advance if everyone makes an effort on their own. For this reason, and even if it is not possible to work on every goal in every project, they have to be carefully analysed one by one. Whatever action it is possible to take in each one is going to be taken, improving the social and environmental impact of the project as much as possible. The 17 goals are the ones shown on the Image 6 and when the initial approach of the project is being made, is really important to take them into account so at the end project a real step forward is achieved environmentally and socially talking.

Image 6: SDG goals *[11]*

On this specific project, the following goals have been selected as priority even if the others are not left aside. This selection has been made considering that the impact of the results obtained are going to be higher on these specific areas, and because by concentrating the efforts into those goals the project ends up having better results on this area. The goals in which the project focuses more and the actions that are taken in each one are the ones stated on the following Table 1:

Table 1: Expected results on the SDG

SDG	Actions to be taken
	To try to fight against poverty, this project is going to generate new job positions. The advance on assembly processes that the project will bring, will lead more enterprises to use this 5.0 industry method and include more workers to collaborate with robots. Therefore, enable more people to access a stable job and a good salary that help them have a better life quality with a higher economic power.
	The collaborative robots and the 5.0 industry also ensure the well-being of the workers before the performing task itself. So, this added to the compliance of all the needed ISO safety measures, the wellbeing of the operators is ensured. Furthermore, creating an ergonomic working station where the robots are in charge of physically demanding tasks, will also improve the operator's heath conditions.
	The implementation of this new assembly method ensures a decent quality product, increasing the enterprises standards and its economic abilities. This directly affects in the working conditions of all the workers on the plant.
	This project on its own is an innovation project that helps the industry advance. It brings new innovative assembly methods that are not used to often on the actual industry and can suppose a big step forward on this area, by bringing more effective and efficient methos to a part of the industry that still remains left behind with respect to other areas.
	Applying an error detection system in the camera reduces the defective assemblies /elements, decreasing the consumption and making the production more effective. The machine is also intended to be clean on its production, not producing any waste. Also, as the energy consumption of the machine is going to be measured, this will lead to adaptations on the process that can minimize this impact.

1.8 State of the art: General project

In the context of Industry 5.0, the focus is on closer collaboration between humans and machines through personalization, sustainability, and creating smarter, human-centred work environments. For a manual mechanical assembly, operator assistance technologies can include:

Table 2: Different industry 5.0 technologies

Technology	Description	Benefits
Collaborative Robots	Robots designed to work safely alongside humans without safety barriers. They assist operators in repetitive or precision tasks, reducing fatigue and improving efficiency.	Improve in productivity and reduction of injury risks, without completely replacing humans.
Exoskeletons	Portable mechanical devices that help workers perform heavy physical tasks, such as lifting or manipulating objects. They can be passive (providing support only) or active (with motorized assistance).	Increased physical endurance and reduced fatigue and injuries, allowing workers to perform physically demanding tasks for longer and more safely.
Augmented Reality (AR)	Uses smart glasses or mobile devices to overlay digital information on the operator's physical environment. They display real-time, step-by-step instructions, indicate specific points where tasks should be performed, or verify correct assembly.	Improves accuracy, reduces errors, and accelerates operator learning, as well as facilitating predictive maintenance.
Voice Assistants and Artificial Intelligence	AI systems can be integrated with voice assistants to provide real-time instructions, help resolve assembly issues, or manage inventory and resources needed for tasks.	They allow operators to stay focused on manual work without having to consult manuals or screens, improving productivity.
Machine Vision Systems	Advanced cameras with artificial intelligence can verify assembly quality, identifying errors or defects before the product moves to the next stage.	Greater quality control without requiring extensive manual inspections, reducing human error.
Wearables for Wellbeing	Wearable devices such as smart watches or wristbands that monitor the health and wellbeing of the worker (such as heart rate, temperature, posture), alerting them to potential physical risks or signs of fatigue.	They help prevent injuries and promote a healthier work environment, aligned with the Industry 5.0 principles of human wellbeing and sustainability.
Machine Learning Systems	Software that learns from human interactions to optimize production processes. For example, it can adjust the pace of the assembly line based on the individual capabilities of the operators.	Improves work customization and helps optimize human-machine interaction by adapting to the needs of the operator.

1.8.1 Selection criteria

In order to select the technology in which the developed product will be based on; some criteria have been considered:

- Price tag: Usually, the principal factor for a company when they think about performing an investment, especially when talking about SMEs with limited budgets.
- Range of application: The selected technologies must be used to create a product that serves a wide range of companies from different industrial sectors. Therefore, the difficulty of implementation and the flexibility when talking about performing different tasks is also a key aspect to consider. Also, the easier it is to implement, the less work will it need to put the product into service, lowering its price tag as consequence.
- Availability of the material / Research cost: The environment where the project is carried out already has some of the previously mentioned technologies installed and ready to use in the laboratory. In consequence, the research will strongly depend on the usage of the available materials.
- Compatibility with other technologies: The listed services and products are not exclusive between each other; they are either complementary. However, some are easier to combine than others, which again influences the implementation and finally, the price tag. The compatibility is scaled from 0 to 7, meaning how many technologies the analysed one can be compatible with.

1.8.2 Comparison table

Table 3: Comparison table

Technology	Price tag	Range of applications	Material availability	Compatibility
Collaborative Robots [22]	> 20000€	Wide	Available in laboratory	6
Simple Exoskeletons [12]	2000-3000€	Low	Not available	3
Complex exoskeletons [14]	40000 – 120000€	Wide	Not available	
Augmented Reality (AR) [15]	3500€ + developing costs	Very wide	Available in another department	7
Voice Assistants and Artificial Intelligence [16]	40000 – 100000€	Very wide	Available commercially	7
Machine Vision Systems [17]	5000 – 20000€	Medium	Available in laboratory	5
Wearables for Wellbeing	1000€	Full range	Available in another department	7
Machine Learning Systems [18]	30000€	Medium	Available in laboratory	4

1.8.3 Final selection

For this project, the most suitable technologies that have been identified are the collaborative robots and exoskeletons (remind that the goal is to generate a technology that can be applied to any sector of industry). These two technologies can be used in nearly any assembly line, while having a significant impact on the productivity and injury prevention of the operators.

In the case of the exoskeletons, their selection varies strongly depending on the type of movement it is desired to help the operator with, as it is designed according to that one specific activity or movement that the worker must perform. This means that either many designs for specific movements may be needed (each product being cheaper), or a unique, more advanced and costly design is done to help with most of the movements done by the operator.

Exoskeletons meant for simple tasks like box lifting are the most common ones in the market and cost around 2000€ and 3000€ *[14].* The price can vary depending on the forces applied to the exoskeleton. Anyway, as the complex exoskeletons are too overpriced and the simple ones have a too limited range of application, neither option seems really viable for this project that intends to reach a big market of enterprises from a wide range of sectors that use different types of movement each.

Even so, they are really useful, and a lot of enterprises are keen to use them, so by means of creating a big variety of simple designs or a good complex one (that would take a lot of time and effort) it can be a good thing to work on.

Image 7: Types of exoskeletons *[12, 13]*

In contrast, the collaborative robots have a much wider range of applications for which they can be designed for. The counterpart, though, is that the price for the robots depends on the size of them rather than the simplicity of the application. For example, the smallest Cobot of Universal Robots, UR3e, costs something around 20.000€, and can hold materials for up to 3kg with a working area of 500mm^3 *[22].* Another advantage of the collaborative robots is that their application can change without needing to replace the robot itself. Just performing an easy and fast reprogramming to the robot, the system could continue working perfectly.

In addition, the research centre in which the project is done is the *Department of Systems and Automation Engineering* (DISA) of the *UPV*. Its facilities are mainly composed of laboratories for control and robotics, with the department nowadays being mainly focused on robotic applications in industry. This is another aspect to consider when talking about the available materials for the project. Specifically, the department had three available robots by the time the project started: ABB IRB140, ABB IRB1100 and two UR3e (Universal Robots). Even though actually only the last one is a collaborative robot, for the reason of this project, the three of them can be used.

ABB IRB140 ABB IRB1100 UR3e

Image 8: Available robots in DISA laboratory *[19,20,21]*

For these set of reasons, the team has decided to work on the collaborative robots. Furthermore, these collaborative robots can also work alongside other different technologies that help on developing an even more advanced workspace that relies more on the industry 5.0 integration. As the implementation of more of these technologies will only improve the process' performance; deepening its control over the process and widening its range of possible applications, some of these technologies have been analysed to know which one of them would add more value to the project. The most common options to combine with the collaborative robots are the following:

- Machine Vision Systems. A camera can be used to monitor the state of the assembly and identify any errors that the different elements can have. This way, the robot will be able to reject wrong pieces and stop the production. It is the most commonly used combination with robots in industry.
- Machine Learning Systems. Usually accompanies Machine Vision Systems. The ML algorithms can be used to teach the robot how the assembly should be done or adjust the thresholds depending on the environment light that creates disturbances in the analysed images. Basically, it is used to further improve the achieved results through Machine Vision Systems.
- Artificial Intelligence. It exists the possibility to implement and AI that can recognize a 3D assembly and generate the required assembly process from it. Meaning that it would be enough to provide a 3D CAD assembly to the robot to start the assembly process. Using Artificial Intelligence one unique product solution may be applicable to any kind of assembly. Creating it is very challenging though and would require deep knowledge about AI generation.

Due to lack of knowledge in the field of Machine Learning and Artificial Intelligence, the project team has decided to focus on Machine Vision Systems. At the moment of the project definition, there is no available camera in the robotics laboratory, so a market analysis is done to find the most suitable camera type and model for this specific application. Finally, to select which of three robots to use, an in deep comparison is done. As this part of the project focuses on the vision part, the next section contains the state of the art for the computer vision, and the one concerning the robotics is located in Annex G.

1.9 State of the art: Computer vision

In this section of the state of the art, the specific market analysis of the machine vision system of the project is held. To accomplish this purpose, and as there is no background on the project about this topic, all the different computer vision systems are analysed and a specific camera that best suits the project will be chosen from all the competitors in the market. But before doing so, first we need to understand how a 3D camera works and what purpose it serves.

The 3D cameras are cameras that not only capture a 2D images like the usual cameras, but they also measure depth by means of different techniques. Afterwards, this depth information is used to create a 3D point cloud map that takes into account the distance and position with respect to the camera of each of the generated points. This point cloud is then used to project the image on it, creating a more realistic interpretation of the surroundings and helping the computer vision system gain extra vital information of the environment (compared with a typical 2D camera), that is crucial in some applications such as this project.

1.9.1 Types of 3D cameras

As previously mentioned, there are different techniques to measure the depth of the image. This used technique can be considered as the main differing factor between 3D cameras, making it the main factor to classify them. Each one of the methods serves some specific purposes better and have special characteristics that makes them more suitable for some specific applications. For this reason, all the main types are individually analysed and compared so that the most suitable one for this project is selected.

The actual market corresponding the 3D computer vision is basically composed of three types of 3D cameras, each one of them explained on the following sub-sections:

1.9.1.1 ToF cameras

As the name suggests, these cameras determine the distance to an object by measuring the time taken for emitted light to reflect off the object and return. Therefore, we can say that unlike the others, they take a time-domain rather than a spatial-domain approach.

The exact functioning of this cameras can be easily explained by looking at the Image 9. Here it can be seen how an IR emitter sends an IR wave (indicated in red) that reflects on the target object and comes back (indicated in blue) to the sensor. By measuring the phase shift between these two IR waves, the distance to the object can be calculated. To calculate this phase shift, the relation between four different electric charge values is considered. The four phase control signals have a 90° phase delay between them, and by measuring the electron collection (electric charge) of each of them with the IR sensor, we can calculate the phase difference ($\Delta\varphi$) between waves: *[22]*

$$\Delta\varphi = \operatorname{atan}\left(\frac{Q_3 - Q_4}{Q_1 - Q_2}\right)$$

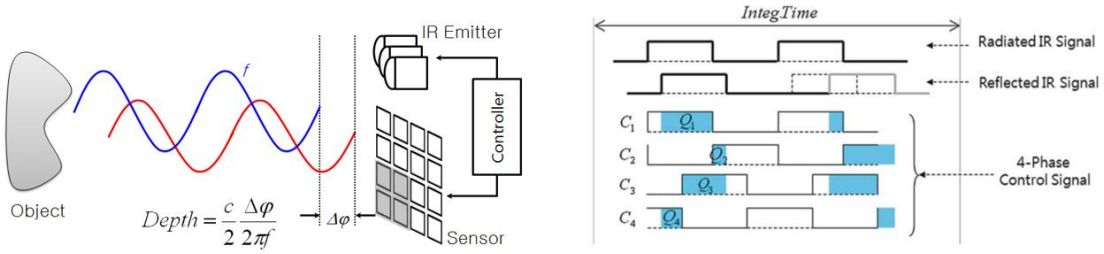


Image 9: Principle of ToF cameras *[22]*

Once having calculated this phase shift, the corresponding distance can then be calculated using the speed of light (c) and the signal frequency (f):

$$d = \frac{c}{2} \cdot \frac{\Delta\varphi}{2\pi \cdot f}$$

1.9.1.2 Stereo cameras

Mimicking human vision, the stereo vision technique uses two cameras displaced from each other to record the same 2D view taken from two different angles. Knowing the fixed relative positioning of both cameras and their angle, a software compares corresponding points in the two flat images, identifies the disparities and through triangulation produces the full 3D point cloud. This functioning method can be better understood by seeing the Image 10.

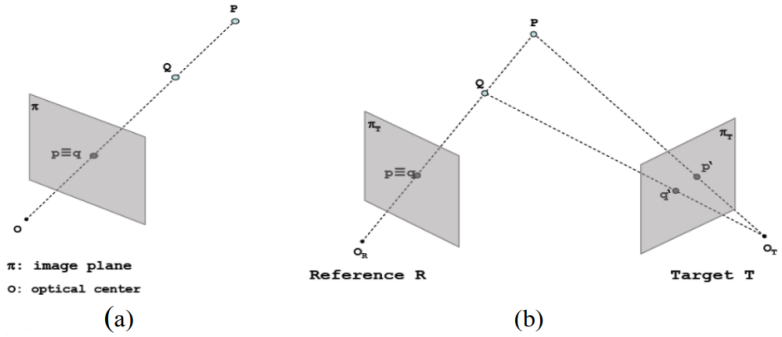


Image 10: Principle of stereo cameras *[23]*

As we can observe in the image from the single camera (a), all the points into the same projection line are in reality the same image point. Therefore, both real points (P and Q) from the target camera project into the same image point (p ≡q) of the reference camera. This occurs for each point along the same line of sight and can be repeated for all lines of sight. So, if the same corresponding (homologous) points of both images are found, triangulation can be used to infer depth. This process can also be extrapolated to a higher amount of cameras and their corresponding images for a more precise result. For this process to be accurate, the most important restrictions to take into account when taking a pair of stereoscopic pictures are the following: *[23]*

- Cameras should be horizontally aligned.
- The pictures should be taken at the same instant.

1.9.1.3 Structured light cameras

Structured light cameras scan the environment by employing a single light source that projects multiple lines typically in the infra-red (IR) spectrum onto objects and later observing the distortion by means of tracking these lines with one or more cameras.

The light sources emit a series of meticulously designed light patterns that are projected onto the object under measurement. During this process, the cameras (placed at a known distance from the projector) concurrently capture a sequence of images of the illuminated object. As these captured images undergo distortion on the surface shape in comparison with the flat surface used for calibration, geometric triangulation can be used to determine the XYZ coordinates of each point of the scanned object's surface.*[24]* This working principle of the structured light cameras can be easily understood by looking at the Image 11.

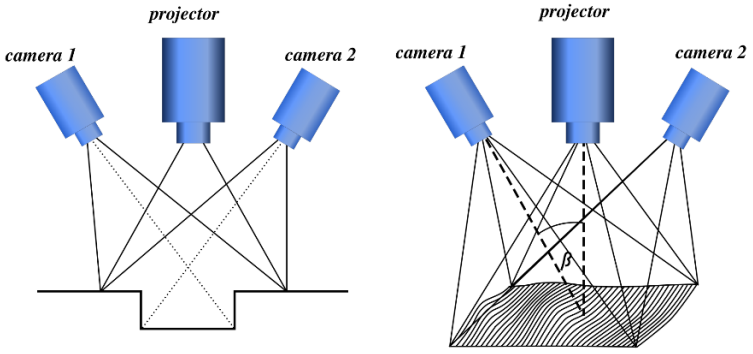


Image 11: Principle of structured light cameras *[25]*

1.9.1.4 Comparison

Each of the three methods have their advantages and disadvantages that makes them more suitable for some specific applications rather than for others. For this reason, and in order to find the most suitable method for this project, the main aspects of each of the types have been compared on the Table 4.

Table 4: Characteristics of the distinct types of cameras

Criteria	ToF	Stereo	Structured light
Price range	Moderate - High	Low - Moderate	Moderate - High
Measurement range	Medium to long (+10m)	Short to medium (<10m)	Short to medium (<5m)
Depth precision	Millimetre to centimetre accuracy	Millimetre accuracy at close range (Declines with distance)	Sub-millimetre - millimetre accuracy at close range
Image quality	Moderate (Often lower resolution than RGB)	High (Depends on RGB camera quality)	High in close range (Declines in high ambient light)
Frame rate	High (<60 FPS or more)	Moderate (<30 FPS, limited by processing time)	Low - Moderate (10–30 FPS)
Ambient light sensitivity	High (Works well in low and high light condition)	Low - Moderate (Struggles in poor lighting)	Highly sensitive (Struggles in high light)
Power consumption	Moderate - High	Low - Moderate	Moderate
Typical applications	Robotics, automotive, outdoor 3D mapping	Close-range 3D applications, electronics	High-precision tasks, close-range scanning

For this specific application, Stereo Vision cameras are the optimal choice out of the three types. Generally speaking, they offer a balanced combination of accuracy, versatility, and cost-effectiveness that suits perfectly on the project. In particular, ToF cameras were casted out as a viable solution due to their limited depth accuracy and relatively lower image quality. While ToF cameras perform well in terms of frame rate and long-distance depth measurement, they lack the high precision required for close-range applications. Their centimetre-level accuracy would be insufficient to reliably identify small chair components or detect minor assembly errors. Furthermore, ToF's broader and less detailed depth capture means that it cannot accurately discern the fine spatial details of the objects or evaluate assembly precision, both of which are essential to the successful execution of this task.

On the other hand, Structured Light cameras could also be an option for this case as they also offer a high accuracy. However, they are less suitable in dynamic environments where continuous positioning and real-time processing are required. Furthermore, they are also more sensitive to ambient lighting and require more controlled settings to function effectively. Additionally, Structured Light systems tend to operate at lower frame rates, which could slow down real-time data processing, especially when fast, consistent depth measurements are needed. Stereo Vision cameras, by contrast, provide high spatial resolution and accuracy in close-range applications without needing strict lighting control or costly additional hardware, making them a more balanced choice in terms of adaptability and cost.

Even though, while Stereo Vision is the most suitable option, there are still challenges that need to be addressed for optimal performance in this application. Stereo Vision cameras rely heavily on good lighting and texture for effective depth estimation, which may present difficulties if parts lack distinct surface features. To address this, additional lighting may be required, which may present difficulties if the parts have smooth, uniform surfaces or lack distinguishing features. Stereo Vision systems also have higher computational demands than the other options, as depth estimation requires powerful processing resources. Therefore, ensuring that the processing hardware can oversee these demands in real-time will be essential to achieve the speed required in the robot's assembly and verification tasks.

1.9.2 Camera selection

Having established that Stereo Vision cameras are the most suitable option for this application, the best specific camera model of that type need to be selected now. Given the critical requirements for high precision, clear image quality, and adaptability in various lighting conditions, the available stereo camera options on the market need to be analysed based on the following criteria:

- **Depth Accuracy and Measurement Range**:
  It is essential that the selected camera provides millimetre-level accuracy at a range of approximately 2 meters to ensure reliable identification and positioning of small components.

- **Image Quality**:
  High-resolution imaging is crucial for detecting fine details on objects, enabling effective recognition and quality control during assembly processes.

- **Frame Rate**:
  A camera with a high frame rate ensures smooth, real-time performance, allowing for quick adaptations during dynamic operations.

- **Environmental Adaptability**:
  The camera must perform well under varying lighting conditions, as ambient light can affect depth perception and overall accuracy.

- **Integration and Compatibility**:
  Ease of integration with existing software and hardware platforms is vital for ensuring efficient deployment and system functionality.

Considering these criteria, several stereo cameras have been analysed. From all the options the *Stereolabs ZED 2i*, *Orbbec Gemini 2* and the *Intel® RealSense™ Depth Camera D435i* were the best options. But after a thorough evaluation, the *Intel® RealSense™ Depth Camera D435i*, shown on the Image 12, stands out as the optimal choice for the project due to the following reasons: *[26]*

- **Exceptional Depth Accuracy**: The D435i provides precise depth measurements within close ranges, making it highly suitable for accurately identifying and positioning small chair components.

- **High Image Quality**: It delivers high-resolution RGB images, which are crucial for distinguishing fine features and enhancing object recognition algorithms.

- **High Frame Rate for Real-Time Processing**: The camera supports high frame rates, ensuring responsive and smooth operation essential for the dynamic nature of robotic assembly tasks.

- **Integrated IMU**: The built-in inertial measurement unit (IMU) enhances depth accuracy by compensating for any motion, ensuring reliable performance in dynamic environments.

- **Seamless Integration**: Its compatibility with various processing platforms and support for popular software libraries streamline the development process and facilitate easy integration into existing systems.

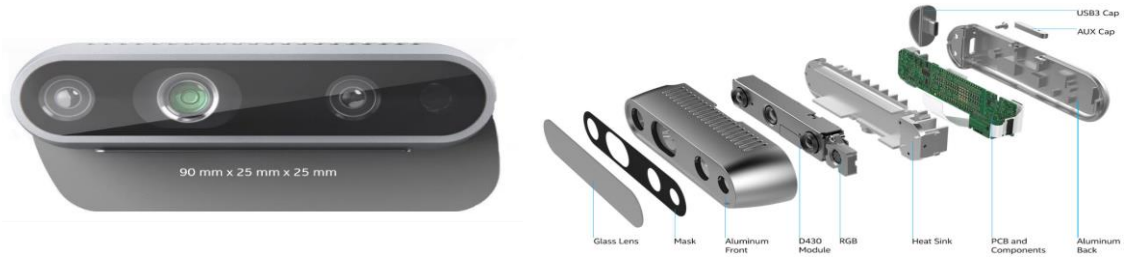


Image 12: Intel® RealSense™ depth Camera D435i *[26]*

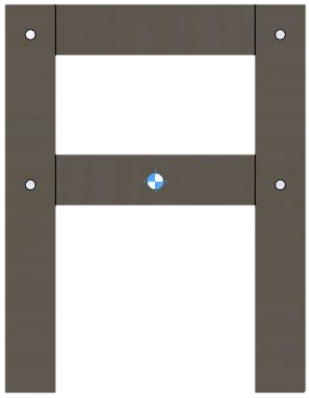
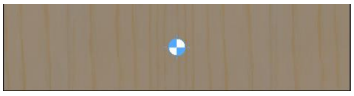
In summary, the *Intel® RealSense™ Depth Camera D435i* not only meets but exceeds the requirements established for an effective stereo vision on this application. The combination of depth accuracy, image quality, responsiveness, and integration capabilities make it the ideal choice for identifying, positioning, and verifying the assembly of small chair parts in a reliable and efficient manner.

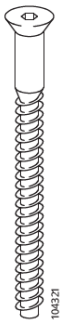
2. Assembly process definition

As this project is developed in the context of Industry 5.0, the cell must not fully replace the operator that was previously doing the same job. The application must improve the productivity as well as well-being of the operator, basically designing the robot to carry out the heavy load applications and improving the ergonomics of the tasks that the operator will need to do himself.

Therefore, the first task in the project development is to determine which activities will each involved part do, based on the assembly instructions by IKEA (see annex E). As the original assembly process is defined for a fully manual environment, one main adjustment has been made to adapt it to the robotic cell. The following elements shown on Table 5 are the ones that are going to be present on the assembly:

Table 5: Assembly elements

Basic elements		
Leg	Finger	Base
		

Connecting elements			
Bolt (code 104322)	Dowel (code 101350)	Bolt (code 122925)	Square (code 122620)
			

From this point on, the elements will be named as indicated in Table 5.

The adjusted process follows as such:

1. Element number 122620 is inserted in the "Leg" by the robot. It is made the first element of the assembly to avoid hitting the "Finger", as the reach of the robots makes it impossible to assemble element 122620 without hitting an assembled finger. All squares are assembled in this stage, first to one leg, and then to the other one.
2. One of the legs is placed in a free area for later use and the other one remains in the assembly space, to carry out the assembly of the fingers. For so, one dowel is inserted into the finger before putting it in the base. The second dowel is put after the finger is assembled in the base. Process is repeated until all finger elements are placed. All this can be seen on the Image 13.

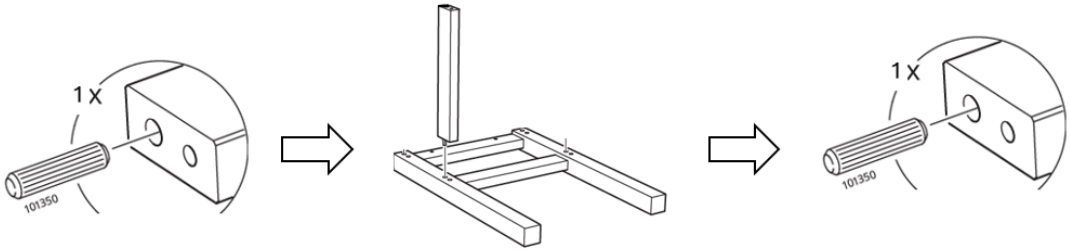


Image 13: Finger-Base assembly

3. Once all four fingers and its dowels are placed, the leg left in the free area will be mounted on top of the assembled structure. Next, the assembly is moved to the free area and oriented towards the operator's position to screw the bolts (code 104322). The operator is then asked to do the screwing as seen on Image 14, avoiding the design of a new tool.

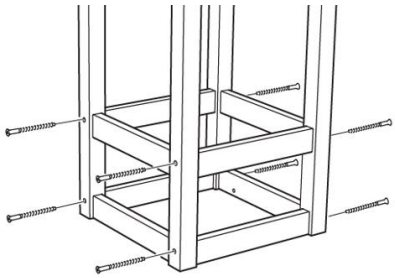


Image 14: Bolt position

4. Finally, the base of the chair is placed in the free area and the already assembled structure is mounted on top. In that position, the operator will need to screw the final bolts (code 122925) as seen on Image 15 and then check that the entire assembly is correct.

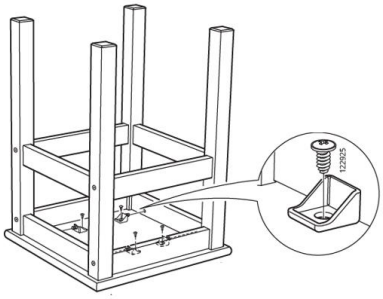


Image 15: Chair base assembly

This process leaves a final task distribution, which's resume is shown in Table 6.

Table 6: Task distribution

Operator	Robotic system
Material supply	Identification of the assembly material
Bolt threading	Element 122460 assembly in base
Monitoring of the robotic system	Dowel insertion
Product testing and verification	Element assembly (without bolts)
	Orientate assembled elements towards operator for screwing accessibility

Nevertheless, the process can be subjected to changes if any issue shows up while testing or designing the control program.

3. Camera information

Before starting to process the information received by the camera, is important to understand how it works. It is essential to know its characteristics, tools and limitations in order to get the most out of the camera. Only by understanding this it will be possible to obtain an outstanding performance when processing the received images.

3.1 Camera selection

As said on the Section 1.8 (State of the Art), the *Intel® RealSense™ D435i* is one of the best options in market for this application, or at least that was the main idea when this evaluation was done. But later in time, the department bought the *Intel® RealSense™ D455f* camera from the same family, so it was decided to compare both and decide whether this new camera was better for this application.

To make the choice between Intel RealSense D435i and D455f for the project, is needed to focus on the crucial factors related to the application. The lighting condition is always in flux, and hence the selected camera should manage variable exposure conditions to ensure depth and RGB images remain stable over varying lighting conditions. Since there is only one camera, multi-camera synchronization has no relevance, but depth accuracy and minimal distortion between depth and RGB images will be critical. It is also necessary to minimize noise, as great depth noise might affect severely object recognition.

The recognition distance on this application is of approximately 1-1.2m for object detection and minimum camera distance for hole identification. Therefore, the operating range, even if images have to be processed at different distances, would fall in the bracket of 1-1.2 meters, which is more crucial with respect to this application. Nevertheless, the important thing here will be the capability of the camera to perform at both distances, focusing on the longer one. So, though we also take the minimum distance into consideration, the most prominent issue is the accurate depth perception around 1m. And in order to compare that the following graph has been generated, checking the RMS error at different distances in both cameras at a resolution of 640x480:

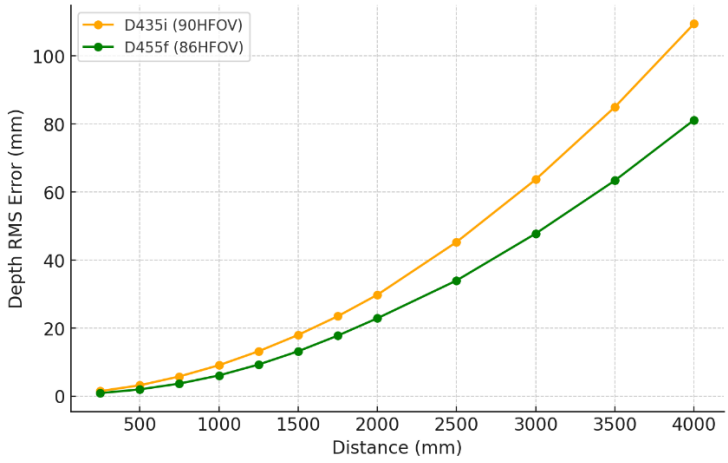


Image 16: Distance error comparison

Besides that, distortion reduction is a critical concern, especially about RGB-depth alignment and depth accuracy. Other factors, such as mechanical design, power consumption, software support, and IMU support, are still valid but relatively weaker in our decision.

Additionally, some terms such as Broadest Vision and IR Pass Vision may need further explanation. In other words, the broadest vision defines the RGB field of view, hence how much of the scene the camera captures in standard colour imaging. A larger RGB FOV can provide more scene context, for instance, object detection. On the other hand, IR pass vision refers to how well a camera utilizes infrared illumination for depth sensing. Cameras with better IR performance can provide more stable depth perception in low-light or texture-less environments; hence, cameras with better IR performance are quite ideal for situations where lighting may fluctuate frequently.

Another thing that is quite important to know for this application is the difference between Rolling Shutter and Global Shutter. Rolling shutters scan objects line by line, hence causing distortions in case of object or camera movements. Global shutters on the other hand capture all frame information at once, thus completely eliminating any possibility of motion distortion and providing better RGB-depth alignment. Since this tries to avoid distortion, a global shutter will be highly preferred.

After knowing all this we can directly compare the features of both cameras in each of the fields in order to do a direct and fair comparison to make the best decision for the application.

Table 7: Realsense D435i vs D455f *[27,28]*

Feature	D435i	D455f	Winner
Depth RMS Error vs. Distance	Higher error due to shorter stereo baseline	Lower error, improved depth accuracy	D455f
Distance Range	0.2m – 10m (better for close-range)	0.4m – 20m (better for mid-to-long range)	D435i (closer min. distance)
Performance at 1-1.1m	Moderate accuracy, more noise at this range	Better accuracy, lower noise at 1m+	D455f
FPS & Resolution	Up to 90 FPS at lower resolutions	Up to 60 FPS	D435i (if FPS is critical)
Depth Accuracy	Less accurate at longer distances (1m+)	Higher depth accuracy at 1m+	D455f
RGB Sensor Technology	Rolling Shutter (potential motion distortion)	Global Shutter (eliminates distortion)	D455f
Depth Noise vs. Distance	More noise at 1m+ due to short baseline	Lower noise at 1m due to longer stereo baseline	D455f
RGB FOV	69° × 42° (H × V)	90° × 65° (H × V)	D455f

IR Pass Vision	Standard IR performance	Better IR performance (more stable in changing light)	D455f
Depth FOV	86° × 57° (H × V)	86° × 57° (H × V)	Same
Stereo Baseline	50mm	95mm	D455f
Low-Light Performance	Standard	Better IR handling for low-light	D455f
Power Consumption	~1.5W	~1.6W	D435i (slightly more efficient)
Mechanical Design & Durability	Standard	Better enclosure, improved calibration	D455f
Software Support & Integration	Same SDK support	Same SDK support	Same

For the particular needs and requirements of the project, the Intel RealSense D455f would turn out to be the better choice. This gives much better depth accuracy at a distance of 1-1.1m, with lesser noise and distortion. The global shutter eliminates misalignment between RGB and Depth and reduces distortion. It tackles changing light conditions better owing to improved IR performance. Its wider RGB field of view makes the identification of objects better.

The D435i is only preferable when higher FPS (90 FPS) or a closer minimum distance (0.2m) is strictly required. However, these factors are less critical than the most important requirements that constitute depth perception with high accuracy and low noise at 1m.

3.2 Provided information

Once having decided the camera, it is also important to know what information this camera actually provides. So, as it has been seen on the comparison, the camera has several images or information channels that can be used on the post-processing step. Each one of them will provide a different type of information with a different configuration, so it is important to know what information of each image is more useful for the specific purpose of the application, so that the post-processing outputs can be obtained the fastest possible giving the most accurate and solid response and using the least information possible.

First, we have the RGB camera. This channel acquires coloured images up to 1920x1080 resolution and 30FPS. It helps in finding textures, patterns, and colour, which may be good in recognizing distinct parts of the image from their appearance. However, the channel does not provide depth information and is also heavily affected by lighting conditions. For this case, an RGB camera is useful for the identification of disordered objects or even verifying correct assembly of parts by comparing the visual appearance of the structure against an expected model.

On the other hand, the depth camera outputs a 3D depth map, which is computed from two infrared cameras capturing stereo IR images. These infrared cameras work in tandem with each other, such that the disparity between the images is internally processed to obtain the depth information. This depth map gives the distance of different objects in the camera's view and is critical in finding spatial misalignments, holes, or other geometric features. The main advantage of using a depth camera is that it can work under changing

lighting conditions since it uses infrared light instead of visible light. However, the depth resolution is lower than for the RGB camera, and noise may appear on object edges or on reflective surfaces. In this specific project, the depth camera will be important in things like detecting objects placed above another or detecting holes in the pieces.

Finally, there is also an IMU that includes an accelerometer and gyroscope, providing information on motion and orientation. Such data could be useful in scenarios where movement may need to be tracked or even used to refine depth accuracy during motion. However, given that the setup is stationary, it is not considered to be relevant.

All in all, a combination of RGB and depth data will be the most effective one for this project. While the RGB camera will help recognizing textures and colours, the depth camera will help detecting spatial misalignment. Even so, these channels do not work on their own as they need to be adjusted to the specific setup and conditions of the lab in order to be correctly tunned for this application.

3.3 Functional specifications and adjustments

After knowing which are the information channels from the camera that will be useful for the application of this project, it is important to adjust them in the best way possible by means of changing the different internal variables.

3.3.1 Resolution

As it's shown on the Table 7 above, one of the most critical aspects for this camera that can affect to the project are the minimum depth distance, the resolution and the obtained FPS. So, while it is desirable to maintain the best resolution possible, it is also important to keep a good FPS ratio and the shortest minimum detectable distance possible. Nevertheless, the increase of one's performance causes the decrease in others as it can be seen on the Table 8 below:

Table 8: D455f resolution trade-offs *[29]*

Resolution	Min Z [mm]	FPS (USB 3.1)	FPS (USB 2.0)
1280x720	520	5/15/30	5
848x480	350	5/15/30/60/90	5/10
640x480	320	5/15/30/60/90	5/15/30
640x360	260	5/15/30/60/90	30
480x270	200	5/15/30/60/90	5/15/30/60
424x240	180	5/15/30/60/90	–

Consequently, in order to find a proper balance that is suitable for this application it is important to understand which are the needs. On one hand, we have the resolution as one of the most critical aspects. A higher resolution on an image means a higher number of pixels on the same space and therefore a higher accuracy on the image processing. But on the other hand, this limits the minimum detectable distance, which is also crucial when detecting the holes. Therefore, a balance between both was found at the *640x480* resolution, as it reduces the minimum depth significantly while still maintaining a high accuracy which it is more important for the application.

The FPS are not so important, as the frame rate is quite good for all resolutions. The only clear difference here is the importance of selecting a *USB 3.1* instead of a *USB 2.0*, as the second one loses frames along the way. So, at this resolution a frame rate of *30FPS* has been selected. This is because even if it is important to have as real-timed data as possible, the more time you give the camera to process the information, the more precise it will be. So, selecting this frame rate allows to have an adequate balance between those two factors.

3.3.2 Depth quality

Another key aspect of this project is the depth quality of the camera, as one of its main purposes will be to detect the depth of the different points/objects on the scene and it is important that this is done the most accurate way possible. So, by using the following set of standard metrics based on accuracy, data validity, and temporal stability, the depth quality of the camera can be obtained. *[29]*

- Depth accuracy: Difference for valid pixels relative to a ground truth surface
- Fill rate: Percentage of pixels that have valid depth values.
- Depth standard deviation: Total spatial noise for each valid pixel relative to a best fit plane
- Pixel temporal noise: Total temporal noise for each valid pixel relative to a best fit plane

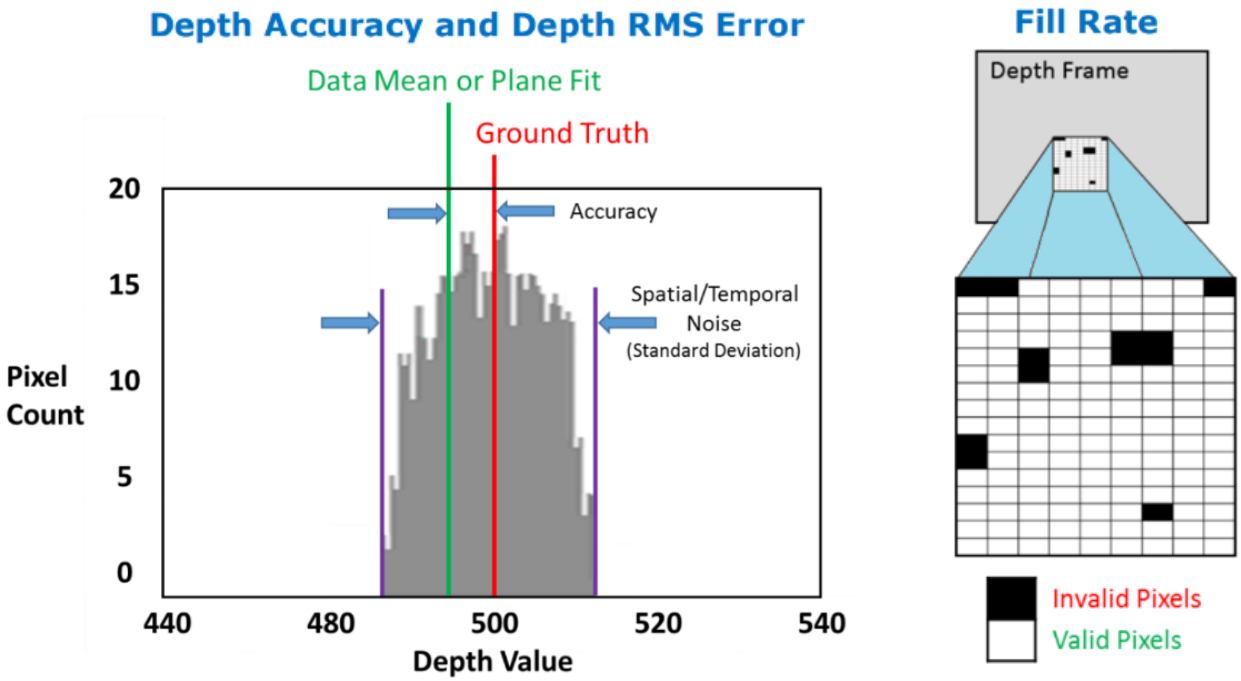


Image 17: Depth quality metrics *[29]*

It is also important to notice that although the modules are designed for a specific depth FOV, the measurements are taken within 80% of this FOV, defined as the Region of Interest (ROI). This ROI aligns with the practical usage area and the module's qualified optical parameters.

So, out of this metrics we can obtain the following depth quality specification of the Table 9 directly provided by the manufacturer for a distance of ≤4m at an 80% ROI and HD resolution at reflect typical conditions on a factory.

Table 9: Depth quality specifications *[29]*

Z accuracy (absolute error)	Fill rate	RMS Error (spatial noise)	Temporal noise
± 2%	≥ 99%	≤ 2%	≤ 1%

Here it is important to note that even if the conditions on which the manufacturer performed the analysis are different from the ones of the project, the difference is not that significant. Therefore, these parameters can be considered as approximated values and can be useful to have an idea of the camera's accuracy even if on the project this will obviously be not as good.

3.3.3 Depth FOV

The depth field of view is also one of the most crucial factors of the camera to consider for this project. It is called depth FOV to the shared overlap of the individual left and right imagers' FOVs for which depth data is provided. Illustrated on the Image 18 for better understanding:

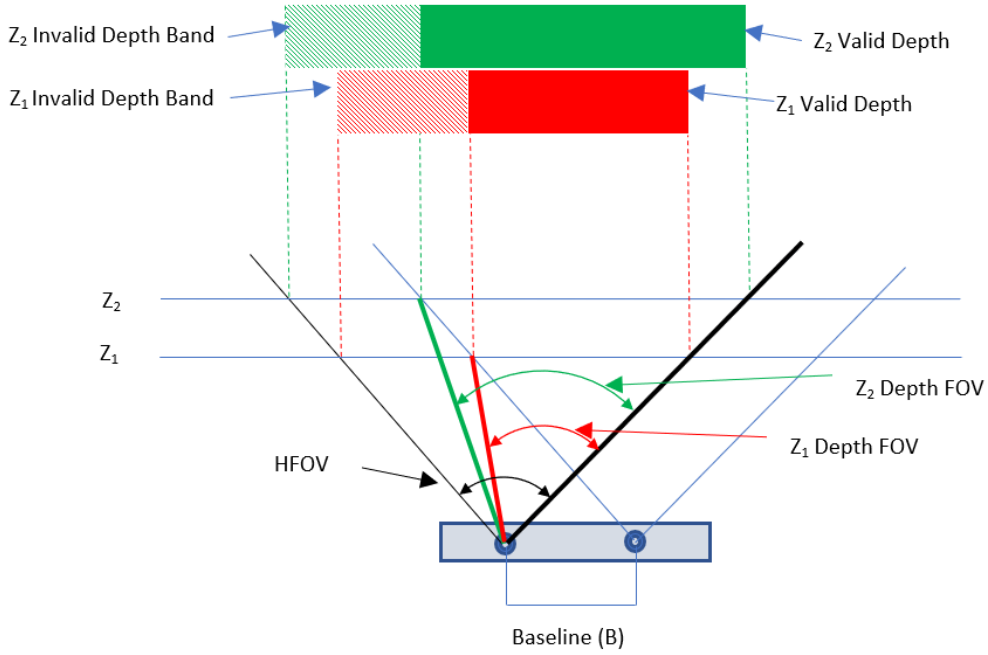


Image 18: Depth FOV *[29]*

And the way of calculating its value for any distance (Z) is really simple, just applying the following equation, where *HFOV* is the horizontal field of view of the left imager on depth module and *B* is the baseline:

$$Depth\ FOV = \frac{HFOV}{2} + \tan^{-1}\left[\tan\left(\frac{HFOV}{2}\right) - B/Z\right] [29] \quad a)$$

Nevertheless, it is important to notice that this FOV changes based on the resolution and aspect ratio. For example, while HD resolution's aspect ratio is 16:9, the VGA's one is 4:3. And their different field of views change as seen on the Table 10:

Table 10: Fields of view values *[29]*

	Horizontal FOV	Vertical FOV	Diagonal FOV
Depth FOV (HD)	87	58	95
Depth FOV (VGA)	75	62	69
Colour camera FOV	90	65	98

Where max. and min. FOV values can vary by a maximum of ± 3 degrees.

3.3.3.1 Invalid depth band

However, the use of this matching algorithm between the left and right imagers, brings an extra problem to obtained image, as on the corner of the image a non-overlapping zone appears as seen on the Image 18. This means that there is an entire region at the edge of the frame that contains no depth data, and this region gets bigger the nearer the depth of this objects to detect is (the smallest Z gets). In this specific case the camera takes the left imager as the reference for the stereo matching algorithm, so the non-overlapped region will always appear at the left edge of the image frame, looking something similar to the Image 19.

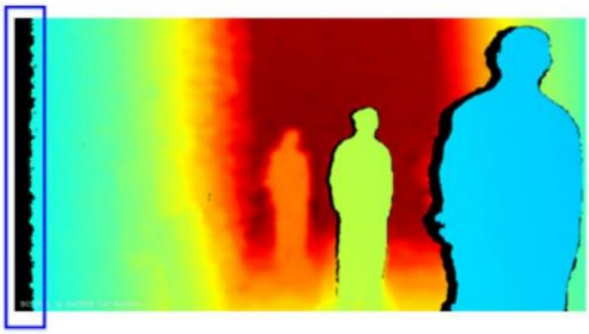


Image 19: Invalid depth band *[29]*

This is something it cannot be changed, so it is important to take it into account at the image processing and also when defining the position where the camera will be mounted.

The width of this invalid depth band can also be calculated in terms of HFOV by means of a simple formula, where HRES is the horizontal resolution:

$$Invalid\ depth\ band\ [px] = HRES \cdot B / \left(2 \cdot Z \cdot \tan\left(\frac{HFOV}{2}\right)\right) \quad \text{b)}$$

4. Working area (Layout)

Before starting any kind of activity regarding designs and programming, it is necessary to define how the elements are going to be placed in the working area. This area is already limited by the tables where the robots are placed in the Robotics laboratory. It has been requested by the promotors not to move them as there are other activities taking place in the same lab and the tables are already distributed with that in mind.

Apart from the tables, some of the robots in the lab are also placed in a permanent position from where they cannot be moved. Only one of the *UR3e* robots can be moved, but it must always be placed in a table end, so that it can be locked in place with a clamp. Also, the base in which the movable cobot is attached to has some premade geometries to place other objects in them.

So, having one of the collaborative robots already in a fixed position, the other must be placed in a position where it leaves enough space for the element placing by one of the robots, and at the same time being close enough to do assembly operations with both robots in a critical zone.

Then, the picking space is defined according to the defined assembly zone, leaving also empty space in case it is required to do some auxiliar operation before the robot enters the critical zone. The picking zone must also be just below the 3D camera for easier object identification and lower distortion from the camera image. However, the camera cannot be placed wherever it is desired, as it must follow the holding structure placed above the robots. Additionally, to avoid light reflections, the entire working space is filled with black cardboard on top of the table.

Considering all these requirements of the space distribution, the working space has been designed in CAD software as shown on the Image 20. This makes it easier to determine the reach of the robots, the picking space and critical zone, also the operator's safe zone.

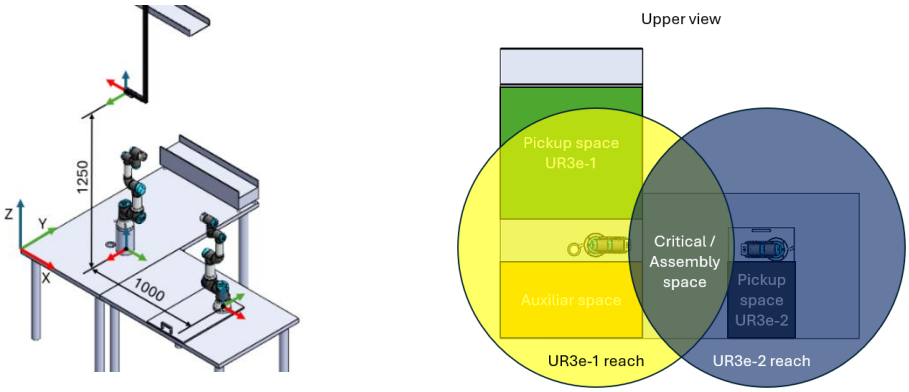


Image 20: Working area final distribution

It can be observed that some areas of the picking space of the first *UR3e* are out of reach for the robot, however, due to the dimensions of the elements, the gravity centres are always in the reaching space, and so, there is no problem for the pickup process.

On the other hand, the critical zone takes approximately one quarter of the working space of each robot, so working with flags and coordinated is primordial. For assembly duration optimization, both robots will be able to move inside the critical zone at the same time. However, this will only be possible if the trajectory of the robots involves creating separation between both tools, in no case they will be able to move simultaneously if in an approaching.

Another security measure for the critical zone is taken so that when the UR3e-1 is holding a wooden element in the air or in a lifting motion, the other robot must always be in a safety position and cannot enter the critical zone.

Knowing the working space of the robots and the security measures taken and given by the cobots themselves, the safety area in which the operator can work is defined. This operator's working area lays between both robots, just in front of the assembly zone and in between the pickup space and the auxiliary storage space. This way, the operator can reach any of the critical spaces and control the mounting process from a short distance while still having a whole open space around him to peacefully perform his tasks.

In addition to all these considerations, some auxiliar tools have been developed to improve the efficiency and speed of the application. All these elements are manufactured with 3D printing, however, as they do not require any tolerances and will not be subjected to stress, the printing method and material was not decided by the team. The manufacturing process was delegated to the department technician.

4.1 Camera levelling

The camera needs to always be parallel to the table where the pickup space is located. Also, it is important to take into account that it needs to be as close to the objects as possible to minimize distortion, while keeping enough distance to avoid collisions with the robot. This is because the precision error of the camera is quadratically proportional to the distance to the objects. Nevertheless, a minimum distance needs to be ensured. Not just to ensure that the robot does not collide against the camera, but also because the camera has a minimum depth measurement distance that will be needed to detect the fingers' orientation with the robot closely placed.

During camera calibration process, it was observed that any kind of disturbance in form of hit or vibration on the supporting rods changes the orientation of the camera in such a way that a recalibration or relevelling is required. But levelling the camera through the support rods takes a lot of time. The rods are quite long and sturdy, so modifying their orientation and position is not easy and requires to operators working at the same time. Also, the bolts need to be tightened to a significantly high torque, generating some bending effect after the tightening, modifying the final orientation of the camera.

Therefore, to make this job easier, a levelling platform shown on the Image 21 has been developed. This tool consists of two parallel plates, one fixed and the other one mobile connected through a four bolt-nut system. Each of the edges of both plates are attached with the bolt-nut system, with the bolt head on top of the fixed plate and the thread side towards the mobile plate. The mobile plate can then be orientated to the desired position modifying the position of the four edges by screwing the nuts in a remarkably effortless way, requiring only one operator. This system also makes it easier to relevel the camera if any disturbance has occurred or a new location is desired.

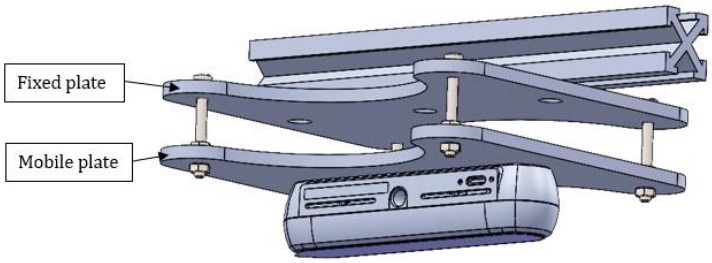


Image 21: Camera levelling tool

4.2 Dowel and bolt supplier

Dowels, bolts and squares are small elements that are difficult to identify with a 3D camera like the *Intel RealSense D455f*. Even the robot tool can have trouble at picking them from the ground. Also, in the case of the dowels and bolts, as they are cylindrical elements, they can start rolling through the picking zone and cause problems with element identification and picking. This problem was considered as highly likely to happen and with huge influence, so a solution was required.

At first, having all elements separated and in exact positions was considered. This requires a significant area and volume though, as all elements cannot be placed in the same level, dowels must be higher, or are at least with enough separation, so that the *FESTO* gripper can pick them with the lower-level dices.

So, instead of separated elements, it was decided to group all elements of the same type and put them in line, so that the first element of each type is always in the same position. Using the effect of gravity, the elements are put in channels with a slope of 45 degrees and roll downwards until they get to the desired position. The dowel and square channels are put in a higher level than the other ones to avoid collisions with the supplier structure. Therefore, leaving the following final design of the supplier shown on Image 22.

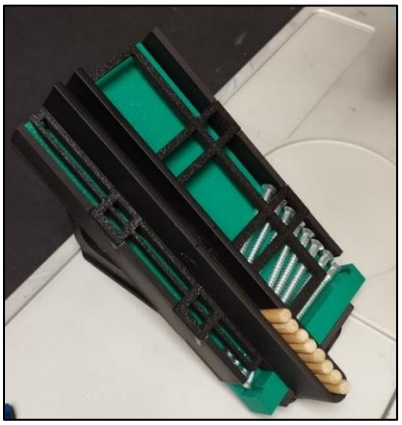


Image 22: Dowel and bolt supplier

This element was then placed at the base of the UR3e-02. This location was selected to avoid placing it at the critical mounting zone, while still maintaining it inside the *FESTO* gripper's reach. Also, it has been taken advantage of a placing space that this base has to keep it sturdy and orient it to the side where it is easier for the robot to pick-up the doubles. Elseway, at this placement the operator has an easy reach to the screws as they lay just next to him and with a correct orientation for their picking.

Also, the bolts are placed on the sides of the dowel channel and in horizontal position for the bolts easier rolling. In the last position, the bolts channels have an opening for the operator's access. Using a magnetized screwdriver and ensuring that the robot is not moving at the moment, the operator will be able to pick the desired bolts by himself. The two bolt channels have openings on top to check the number of elements left in the supplier.

4.3 Final setup

Once testing the real setup started, the artificial vision team realized that the lighting conditions of the laboratory do not allow for proper segmentation of the objects of interest. The windows around the robotic cell and the roof and the glass layer on top of the table generate unavoidable reflections of the light. Furthermore, the light coming from the side wall windows could be in some way controlled by using curtains, but the upper window does not have curtains or any other device that could be used to control the light. This made the working area a non-controlled lighting environment, making it ridiculously hard to adjust the segmentation thresholds.

So, in order to minimize reflections and control the lighting it was decided to use a black cardboard layer on top of the glass layer. This solution eliminates reflections and improves contrast, critical factors for accuracy. Black cardboard absorbs light, minimizing the unwanted reflections. It also creates a dark background that contrasts with most light objects, making image segmentation and analysis easier, as the difference between the object and the background is more pronounced, resulting in more reliable and robust machine vision.

Another adjustment made concerns the separation between robots. At first, this was set at 1000 [mm], but when testing some assembly operations, the robot reached the desired points in singularities or very limiting configurations. The robot did not even allow for the use of the force mode due to the previous reason, essential for the insertion and hole finding. By decreasing the separation by 20 [mm], the problem has been solved.

All in all, the final setup can be seen in the Image 23.

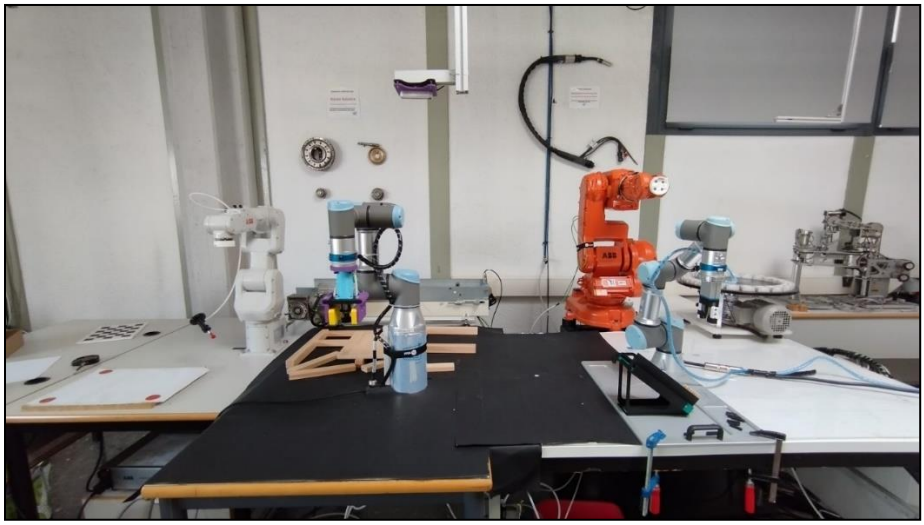


Image 23: Final layout

5. Camera calibration

Once knowing what information, the camera gives, and with that decided how to arrange all the different elements on a layout that fits the mounting process as efficiently as possible, it is time to calibrate the camera itself. This means adjusting the image acquisition process as much as possible by changing things like the like the lightning conditions or the background colour so that the obtained image needs to suffer as less post-processing as possible. All the work done during this initial step of the process is crucial to obtain a clear image that doesn´t need to suffer changes and makes its treatment a lot easier, cutting out time and complexity on following steps.

5.1 Background

The first measure taken for this purpose has been to add a black plane background on the visible working area by the camera. This is one of the easiest methods to add a static background that interferes with the environment as less as possible, allowing the camera to focus just on the essential element and drastically reducing noise.

On one hand, the table where the elements need to be placed was not of a single colour. It was mainly white, but it also had black lines and other different stickers and logos of various tonalities that added a lot of noise to the image. Having a monochromatic background is a clear advantage with respect to having a messy one like before, not only because is less noisy but also because it allows you to just avoid that specific colour's bandwidth and focus on the rest. This is possible because each colour reflects just a specific bandwidth from all the spectrum emitted by the source light as seen on the Image 24. So, if that bandwidth is erased from the captured image, you are left just with the interesting part of the image to analyse.

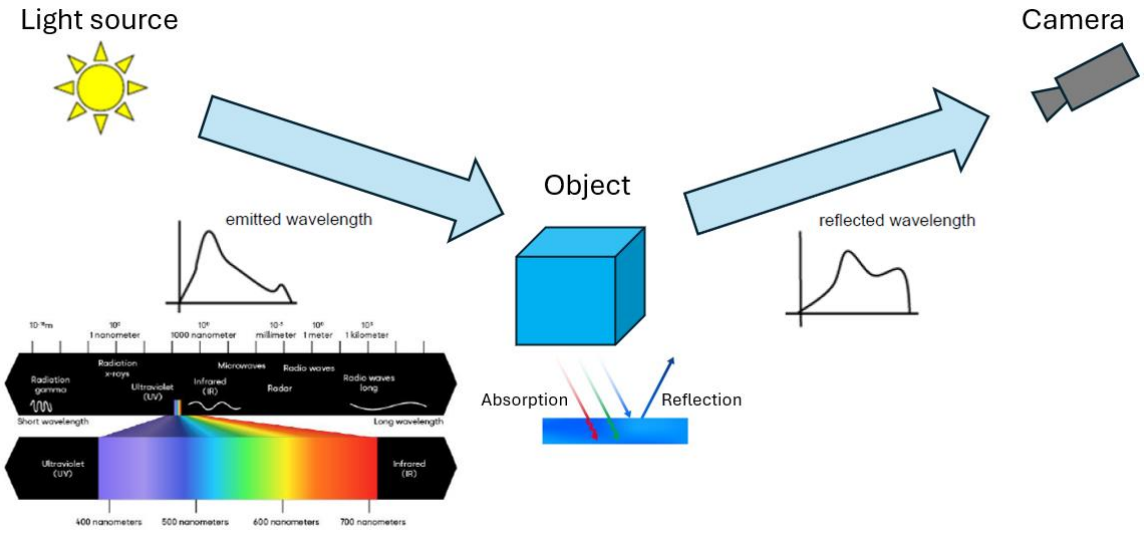


Image 24: Light transmission process *[30]*

Furthermore, the background colour was selected to be black because in that way it is much easier. Not only because it has the highest contrast against the almost with wood of the objects, but because black is a specific colour that absorbs all the wavelengths, so you don't need to erase no colour spectrum and the objects are clearly visible for any information channel that you use on the image processing (RGB, HSV, Gray-scale…). Moreover, as it was previously mentioned, the main purpose of the camera calibration is to acquire the best image and apply the less pre-processing as possible, so avoiding that extra step is something convenient that helps in that objective.

On the other hand, the previous table had a really reflecting transparent plastic layer on top, which created horrible lightning reflection conditions for the image acquisition. This was a problem due to the fast changing and uncontrollable light condition of the lab, which stand far away from the typical industrial environment where the lightning can be precisely controlled. So, as the situation cannot be changed, the best idea is to reduce the adverse effects of the actual conditions by applying a background that offers the most stable and X reflection possible, being a cardboard one of the best affordable options for the project in this situation.

Consequently, big pieces of black cardboard were carefully placed all around the working area, covering all the visible area of the camera. This carboard though, fulfils also another purpose on the same time. It does not only improve the image acquisition of the camera, but it also helps visually limiting the working spaces of the layout making it easier for an external worker to know what the working area is.

5.2 Self-calibration

With the environment and lightning being already defined, camera's intrinsic parameters can start to be calibrated. For this *Intel Realsense* cameras that is an easy task, as the own software the company provides for free offers three automatic calibration modes designed to maintain depth accuracy in varying environmental conditions: On-Chip Calibration, Focal Length Calibration, and Tare Calibration. These modes enable precise depth vision and compensate for potential errors that may be introduced over time due to variations in temperature, mechanical stress, or baseline misalignments.

On-Chip Calibration is a real-time, automated process that dynamically adjusts small depth errors. To achieve this, it detects and compensates for small misalignments or distortions between calculated depth points. It does not use any external hardware, and it is performed directly on the camera's ASIC chip, hence being a fast and effective solution. It continuously monitors and compensates depth accuracy to deliver stable performance in dynamic scenes. This improvement of the calibration can be seen on the Image 25.

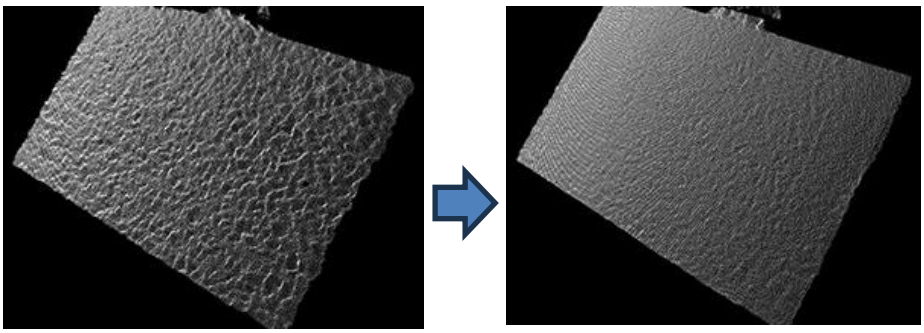


Image 25: On-chip calibration *[31]*

Focal Length Calibration is intended to eliminate depth perception errors caused by optical system shifts or camera distortion. Physical stress, thermal fluctuation, or physical impacts cause the focal length of the lenses to shift slightly, warping depth. Focal Length Calibration quantifies and compensates for the focal length as it can be seen on the Image 26 based on a reference pattern or known environment such that depth perception is consistent and performance is best irrespective of condition.

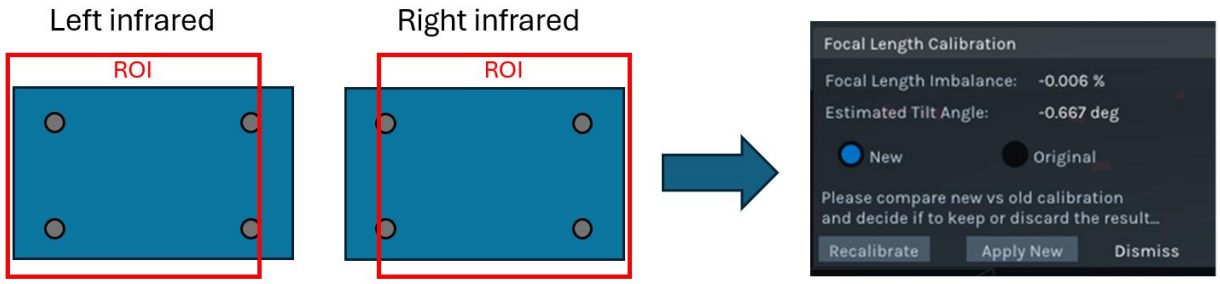


Image 26: FOV calibration

Finally, Tare Calibration compensates for systematic depth offsets by changing the baseline distance between stereo camera sensors. Tare mode requires a known surface to precisely calibrate depth measurements so that depth calculated equals actual-world distance. It corrects internal calibration settings by comparing calculated depth of the camera to a known reference and eliminates depth-measurement bias to provide long-term compensation for depth inaccuracy.

Additionally, the Intel RealSense D455f camera includes a Health-Check function that provides a direct metric of the camera's calibration state. This function evaluates the extent to which the camera's depth accuracy has deviated from its optimal calibration. A Health-Check value below 0.25 indicates that the camera is well-calibrated, while values exceeding this threshold suggest the need for recalibration. If the Health-Check value surpasses 1.0, the camera is considered significantly out of calibration and may require immediate recalibration to restore proper depth accuracy. This function allows users to periodically assess calibration performance without requiring external targets, ensuring the camera remains in optimal working condition just by checking where the health-check value stands on the following scale of Image 27.

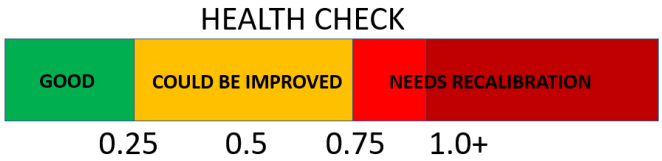


Image 27: Health-check indicator metric *[31]*

Basically, On-Chip Calibration provides real-time correction, Focal Length Calibration helps maintaining optical integrity and Tare Calibration eliminates systematic depth error. So, by using these calibration methods and the health-check function, the performance of the camera has been improved for the application, delivering much more precise depth measurement and a more reliable performance.

5.3 Information transmission

Another key aspect to consider is how the images will be transferred to the computer. This part plays a vital role on the project as the images cannot be processed on the camera itself and need to be transferred to a computer. For this reason, making a proper election on how this frame transmission is done is crucial.

In the general scope of image processing, a lot of information transferring methods exist, each one with its pros and cons. For example, the *CoaxPress,* that is used for high-speed video transmission or the *CameraLink,* that is used when a high bandwidth is needed.

Nevertheless, this exact camera only allows using one single method: the USB. However, even within this technology a selection criterion remains. The camera is prepared to allow USB3.0 cables, but USB2.0 cables are also a possibility. While USB 2.0 can achieve transfer speeds of up to 480 Mbps, USB 3.0 offers a much higher data rate, reaching speeds of up to 4.8Gbps, as it can be compared on the FPS comparison done between both on Table 11. This difference is crucial, as faster data transmission ensures lower latency, higher frame rates, and a more stable video feed, which are essential for the project's image processing requirements. Additionally, USB 3.0 improves power management and overall efficiency, reducing the chances of dropped frames or delayed image transfers. For example, one of both Infrared channels is lost if the USB2.0 is used. For these reasons, USB 3.0 is the preferred option, as it guarantees optimal performance and reliability.

Table 11: USB 2.0 vs 3.0 *[32]*

	USB 2.0	USB 3.0
Transfer rate	480Mbps	4800Mbps
Data flow	1 way	Bi-directional
Picture		

Another crucial factor to consider is the length of the cable and the number of interconnected cables used. To ensure the best possible performance, the cable should be kept as short as possible. Longer cables introduce signal degradation and potential transmission delays, which can negatively impact image acquisition. Moreover, avoiding any unnecessary splices is essential, as these can further compromise signal integrity and create points of failure. Using a single, high-quality USB 3.0 cable without interruptions will provide the most stable and efficient data transfer between the camera and the computer, ensuring smooth and uninterrupted image transmission.

5.4 Coordinate reference

Finally, the last crucial aspect that comes into play on the calibration is the difference between the points obtained by the camera and those exact same points with respect to other bases. So, for example two libraries could have their own coordinate system with their origins placed at different points in space and their axis vectors pointing in different directions. Therefore, each time a point obtained in based on one reference system but needs to be referenced in another coordinate system, a transformation needs to be performed. Therefore, for the calibration is not just important to know these transformations, but also

trying to minimize them is a key factor to improve the performance, as performing these transformations does not only increase the running time of the performance but also reduces the accuracy as the errors can also be amplified when performing these transformations.

Luckily, all the libraries used on python use the exact same coordinate system of the camera, so there is no need to perform any transformations between them. This is because they all follow the standard coordinate system when treating with images; placing the origin on the upper-left side of the image on the pixel (0,0), and with the X axis pointing to the right while the Y axis points downwards. This system differs from the typical coordinate system anyone would intuitively think of, and some other libraries use; of having the origin on the bottom left corner, the X axis pointing to the right side and the Y axis pointing upwards, as it can be seen on the Image 28.

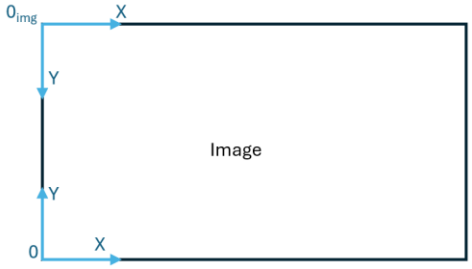


Image 28: 2D differing coordinate bases

This means that all the image processing can be made in a straightforward way, just having to perform coordinate transformations when a point needs to be referenced with respect to the robot's origin instead of the camera. For this purpose, the transformation matrix has been calculated by following a simple process and knowing it has a shape like the one shown on Image 29.

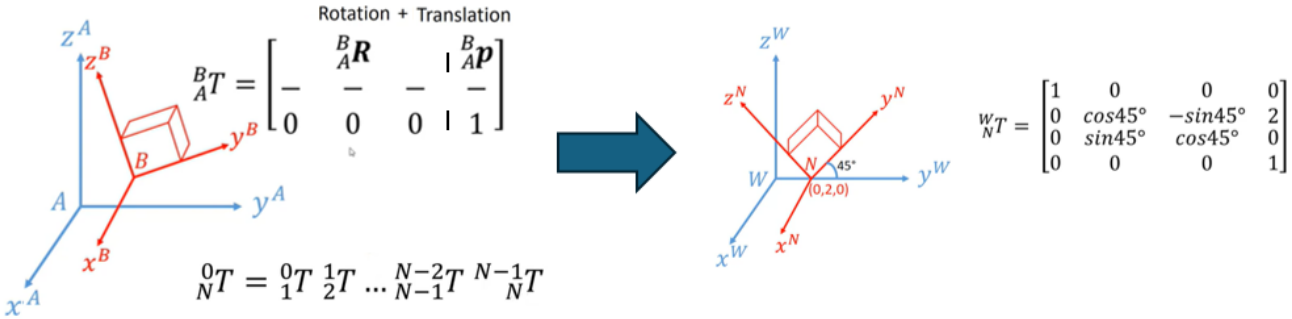
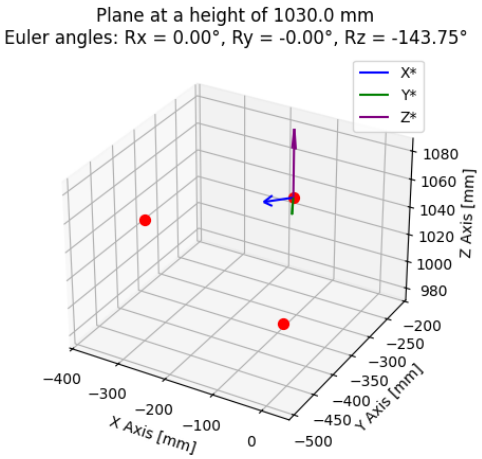


Image 29: Transformation matrix *[33]*

To follow this process, first an object that can be identified by both the camera and the robot was created. This object will basically be a third coordinate system that could be referenced to both the camera and the robot. Then, both tools were individually used to reference the object with respect to their own bases and create a transformation matrix. And finally, both transformation matrices are multiplied to obtain a more accurate camera-robot transformation matrix.

So, to ensure that this intermediate "object axis" can be referenced on the most accurate way possible, the created object is directly a perfect plane composed of 3 points that form an axis. This enables easily and accurately detecting the centre of those 3 points to form an X-Y axis, from which the Z axis can be simply obtained with a multiplication of both other vectors.

From the camera side, this is exactly it has been done. An object detection algorithm is first applied to detect the three points and their centres. Then within these points the central one is selected as the origin and the other two are used to form the X and Y axes. To be able to obtain the Z axis though, the perpendicularity of the axes must be checked, and if this is not the case some little adjustments are applied to match this angle. Once having this base created the rotation angles with respect to each axis and the translation to the origin are calculated to directly form the transformation matrix. All this being shown on the result of Image 30.



$$ {}^{p}_{c}T = \begin{bmatrix} -0.577 & -0.821 & 0 & -350.052 \\ 0.823 & -0.569 & 0 & 185.605 \\ 0 & 0 & 1 & 1030 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Image 30: Intermediate base and its transformation matrix

From the robot's side, a specific tool has also been crated to perfectly measure the exact points. This tool consists of a sharp point that allows seeing perfectly where the robots is placed on the X-Y plane of the tool. Apart from that, as the tool has been designed for this specific purpose and low tolerances, the exact height of the tool and therefore the exact location of the TCP with respect to the robot base can be precisely known. So, thanks to this design shown on the Image 31, the exact location of the points in touch with the TCP of the tool can be known with respect to the robot base.

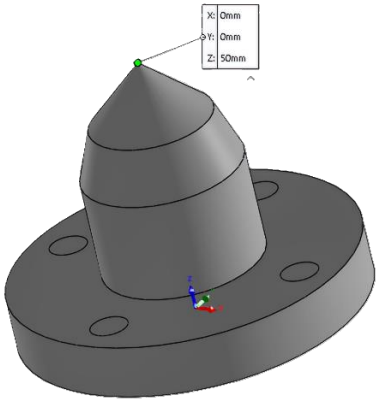


Image 31: Robot calibration tool

By using this tool, the robot was manually placed at the exact same centres of the 3 points that create the intermediate base. Then the exact same process as in the camera is followed to create the intermediate base and the transformation matrix from robot to object is calculated, with the only difference that the points are now given by the robot instead of obtained by the camera. Afterwards, by performing an easy multiplication of both obtained transformation matrices, the homogeneous transformation matrix from the camera to the robot is calculated.

$$ {}^{R}_{C}T = {}^{R}_{p}T \cdot {}^{p}_{C}T = {}^{R}_{p}T^{-1} \cdot {}^{p}_{C}T \qquad \text{c)} $$

The most important thing to consider here is that these points used need to be properly fixed in space and perfectly plane, as a simple movement on those points in between measurements could highly affect on the errors obtained on the transformation matrices from both sides. So, the more stable these points are attached to the space the more accurate the measurements will be.

5.4.1 Accuracy error

This used method does not only provide with a more accurate transformation matrix but also gives the ability to analyse the precision of the camera with a higher precision, as the transformation matrix can be divided in two parts: robot and camera, being able to split the measurement error in both sides. So, as for this case the robot gives an almost perfectly accurate precision in position measurement, it is possible to assume that most of the total accumulated error will be almost exclusively provided by the camera.

Therefore, to measure the error given by the camera measurements, the following methodology has been used: The object used to create the intermediate axis is placed at different points. And for all the different points where it is placed, a ${}^{R}_{C}T$ transformation matrix has been calculated. After repeating this process 10 times, all the different matrixes are compared. Ideally, they should all be identical. But as this is not true, a mean value is obtained to be as precise as possible. Then, once having this mean value, the standard deviation can also be calculated and used as a representation of the accuracy error of the transform. In this case, and assuming that the angular error is neglectable (<0.006), the vectorial space error has been defined as the following:

$$ T_{\sigma} = \begin{bmatrix} 0.006 & 0.002 & 0 & 2.86 \\ 0.002 & 0.006 & 0 & 3.27 \\ 0 & 0 & 0 & 0.74 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow Euklidean\ error\ (X,Y) = 4.34\ mm \qquad \text{d)} $$

$$ \text{where, } T_{mean} = \begin{bmatrix} 0.03 & 0.999 & 0 & 682.75 \\ 0.999 & -0.03 & 0 & 420.73 \\ 0 & 0 & 1 & 846 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

This procedure has been followed keeping all the measured points on the same X-Y plane, but the change in height has also an enormous influence on the measured error. This conclusion can be directly obtained from the information previously explained on the Section 3 (Camera Information) along with the information given by the camera's Datasheet. Nevertheless, in order to quantify and show the true change in error measurement for this specific case, the previously explained steps have been repeated for different heights. Thus, obtaining a change on error base on depth shown on the following graph of Image 32.

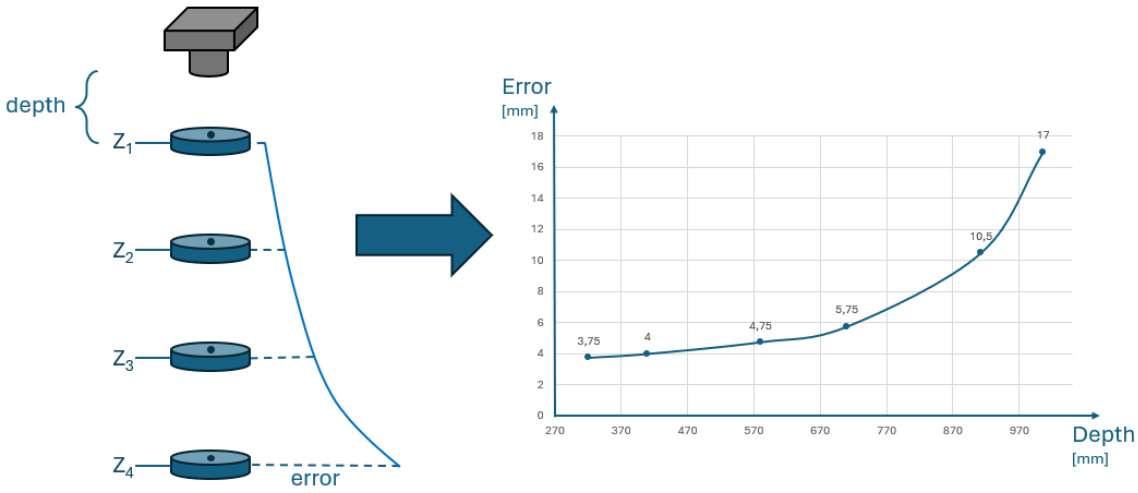


Image 32: Depth dependant error

Furthermore, the repeatability error of the robot has also been measured to know how accurate this is when following a movement instruction. So, on the following Image 33, it can be seen the error committed by the robot each time an instruction to go to the centre points was given to it, repeated 50 times. Clearly, it can be seen that the error here is really small (about 1mm) and that even if this error needs to be considered, is much smaller than the one of the camera.

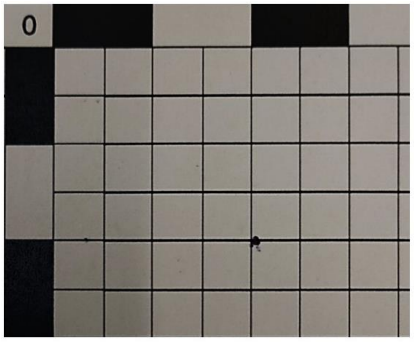


Image 33: Robot's repeatability error

So, as a conclusion from these measurements, we can extract this error to the global one obtained on the transform matrix, concluding that just ±25% of the error was caused by robot inaccuracies and the ±75% of the error left caused by camera inaccuracies. This leaves us with a camera measurement inaccuracy of about ±3.3mm that can cause robot misguidance when moving it to the object picking points.

As this error has already been reduced the maximum possible, but it is still not inside the boundaries of ±2.5mm tolerances accepted by the tool, some extra measures have been taken in the process that ensure correct object picking. But as this has been developed on the robotics part of the project and has no influence on this part, it will not be further discussed here.

6. Communication

For this project, effective communication between hardware components and processing units is essential. The system relies on a 3D camera connected to a local computer via a USB 3.0 interface for high-speed data acquisition. Following initial image processing on this computer, the data is then transmitted wirelessly using a Wi-Fi connection to a separate computer that controls the robotic cell. So, it can be said that communication is the main core of the project, binding all of the components of the project together and becoming the interaction point between the robotic side and the vision side of the project.

6.1 Camera to computer

The initial step in the data pipeline involves the transfer of image data from the camera to the local computer, as the images cannot be directly processed on the camera and need an intermediate interface for this step. This connection is established via a USB 3.0 cable, ensuring high-speed and reliable transmission of the raw image data. A deeper explanation of the logic behind this choice, including a comparison with other alternative transmission methods, was previously mentioned on Section 5.3: Information transmission.

6.2 Computer to robot

The second step would be sending the already processed data to the other computer, which works as a robotic cell. This communication is done via Wi-Fi, and threading structures are used to handle the communication between computers. This wireless link is crucial for providing flexibility and mobility to the robotic part of the project, allowing the deployment in various environments without the constraints of physical cabling. The use of threading ensures that data transmission is efficient and does not interfere with other computational tasks on either computer, maintaining a responsive and reliable connection for real-time robotic control.

But this communication is more useful than just sending some information to the robot whenever the image is processed. This communication thread is the main thread out of all because it has entire control over the camera and the image processing. The local computer is continuously running the threading pipeline, waiting for communication from another device. The robot's computer is then connected to the local one whenever it needs some information. When the connection begins, the robot's computer sends a specific message to the local one depending on his exact needs, and then the local computer operates the camera and the processing depending on the sent command. This message can send 3 different commands:

- **Detect finger orientation:** When the command "finger orientation" is received, the camera takes a picture and the "Hole orientation" function is activated. This happens when the robot has picked up a finger and has already placed it beneath the camera, as it needs to know the order of the 2 existing holes on its end in order to know how to orient it while mounting. This way the exact orientation of the finger is detected, and the position of the holes is sent to the host as a string.
- **Detect objects:** When the command "top elements" is received, the camera takes a picture of the workspace and detects all the elements placed on top. Afterwards, the object list is sent with all the

needed information by the robot as a .JSON file, so the receiving computer knows exactly where it needs to move the robot next to pick up the desired object. This happens when the robot has ended his current mounting task and needs to pick up the next object.

- **Server closing:** When the command "¡Closing server!" is received the local computer ends the threading and shuts up all the communications. This works as a termination command that stops the code whenever the mounting process has ended, and the camera is no longer needed. Anyway, before closing the server, the local computer ensures that no information is being processed. Otherwise, it ends until the corresponding response has been sent and then shuts the server down.

After every communication, a response is sent, and the connection is ended. But the server remains open until the shutdown instruction is sent by the robot's computer, to tell the local one that the job is completed, and the code can be finalized. Otherwise, it would be continuously running waiting for commands eternally. This could be useful if the chair production line would be continuously running and chairs would be produced one after the other. But as for this project this is not the case, the code needs to be ended at some point. Additionally, this stopping command could serve as a safety measure if everything needs to be stopped at some point for reasons of force majeure.

In spite of that, the local computer also replies many other messages corresponding to the possible communication errors. These errors include among other messages that indicate the incorrect format of the received command, the lack of detected objects on the workspace, errors on message readings, unexpected errors when creating the .JSON file, etc. These errors help the host computer having a deeper understanding of the issues happening each time and also giving a hint on how to fix them whenever it is possible.

In order to handle communications in between threads of the same code, global variables need to be used. These variables ensure that whatever result the images give on the processing thread, the communication thread can also read it and therefore know how to act. Thanks to these variables, the communication thread does not only know when the processing has ended to know when to send back the reply but can also access to the sending information itself. This for example would be the case of the "holes position" variable that tells the finger's orientation. But the global variables also work the other way, as they are the key to send the image capturing and processing triggers from the communication thread to the processing code.

Nonetheless, the server can handle various different clients at the same time. This is interesting if more than one robotic cell is wanted to be operated at the same time, leaving a single computer to handle all the cameras for all the different stations. This is not necessary for the project at the moment, but it gives him some scalability, as if in the near future the application is implemented at a real production line it is almost certain that more than one robotic workspace would be operating at the same time to increase production. In that case, having one computer for each camera of each workspace would be inefficient, so having the ability to handle more than one client with their respective cameras from a single computer would create a centralized structure that enables an easier management and flexibility to the system.

Apart from that, keeping the server open for any device also enables the possibility of keeping the local computer untouched on the production line while still having the ability to extract useful information from it with any other computer. So, if some engineer at the factory would want to extract some information, he will only have to connect his computer to the server and request the desired information. This ensures the complete independence of the local computer, avoiding any unplanned changes on the code that could cause small failures on the system's performance. This would lead to a communication interface like the one on Image 34.

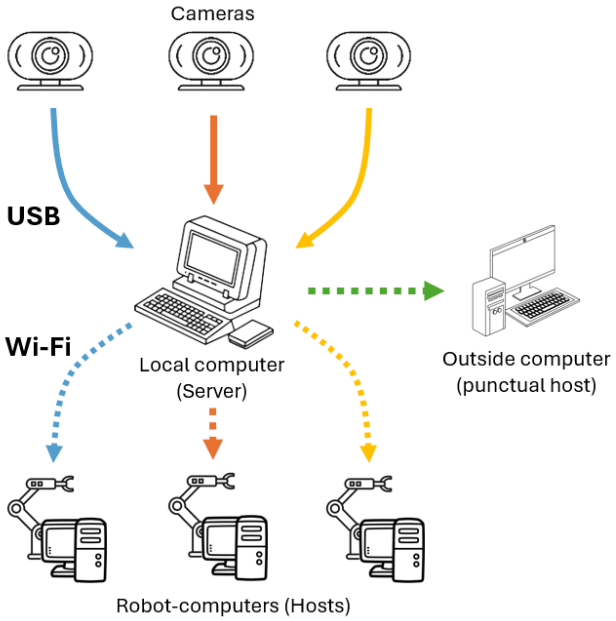


Image 34: Communication interface

7. Programming

With the robust communication infrastructure established, the true intelligence of this project resides within its software implementation. This section will explain the programming architecture that governs the entire system, from the initial data acquisition and processing to the interaction with the robotic cell. As the main core of this project, the program is responsible for translating raw sensor data into actionable commands for the robot, ensuring seamless operation and achieving the project's objectives.

This program has suffered a lot of changes throughout time. In this section only the final code will be explained. If more information about previous functionings and the code's evolution is desired, the old codes and their purpose can be found on the Annex B.

7.1 Main program

The main program, or streaming module, serves as the central control unit for the system. Its core function is to manage all the different capabilities of the code, from the communication with the robotic cell to the acquisition and processing of images from the camera. Previously it operated as a continuous stream, analysing all the images taken by the camera at the specified frame rate. Nevertheless, this module now acts as a command-driven system, capturing images only when a specific instruction is received from the robotic cell, thereby optimizing data flow and processing resources. The specific methods used for communication between hardware elements has been explained on the Section 6 (Communication).

The main program manages the entire vision system's operation, from camera initialization and frame acquisition to dispatching images for specialized processing, all driven by commands received from the robotic cell. It begins by initializing essential components, including the *Intel RealSense* camera with its stream configurations. While the camera is capable of providing various frame types, our project specifically utilizes only the colour and depth frames, as these are the sole meaningful data streams required for our objectives. This, along with many other details such as the stream configuration has been specified on the Section 3.

After that, the critical communication handling system is immediately launched as a server thread in the background. This thread is responsible for handling all Wi-Fi communication with the robotic cell and interpret the received commands. This dedicated communication thread makes the entire system command-driven.

Moreover, to ensure a clean startup and reliable data, an initial set of 20 frames from the camera are discarded. These initial frames are typically noisy and do not provide valid information, so proactively discarding them significantly enhances the safety and accuracy of subsequent image processing.

After that, the image acquisition and processing step can start. But unlike earlier iterations that continuously streamed frames, the current system operates on a command-driven basis. The main loop actively waits for a "capture" order from the robot, signalled via a specific communication event. This ensures that image acquisition and processing only occur when explicitly requested, thereby optimizing resource usage. Upon receiving the command, the program attempts to acquire new frames from the camera, with a timeout mechanism to prevent the system from getting stuck if frames are not available.

Once both frames are correctly acquired, a critical frame validation step is performed to confirm that they are valid and complete. If any frame is missing or corrupted, the current acquisition cycle is skipped, and the robot receives information that no valid data was obtained. If this happens the process will be repeated until a new frame is obtained, but if that does not happen the system I useful for preventing the robot from waiting indefinitely.

Once the frames have been captured, it is important to notice that the raw frames obtained exhibit distortion, and that their pixels do not directly correspond between the RGB and depth images. The depth image, in particular, often appears "holey" or distorted, making direct pixel-to-pixel correspondence unreliable. To overcome this problem, a robust solution for RGB-Depth correspondence is crucial for accurate feature extraction and information sharing between the two image types. For this reason, the program uses *RealSense*'s intrinsic alignment functionality. Specifically, an “align_to_color” operation has been performed, meaning the depth frame is geometrically transformed to precisely match the perspective and pixel grid of the colour frame. This alignment methos has been selected over the “align_to_depth” methos, because colour images generally offer higher resolution and are less prone to inherent distortion. Aligning the depth data to the colour perspective simplifies posterior processing, as both image types then share a consistent pixel grid, allowing for direct correlation of visual features with their corresponding depth information. The comparison between raw depth frame and the aligned one can be seen on Image 35.

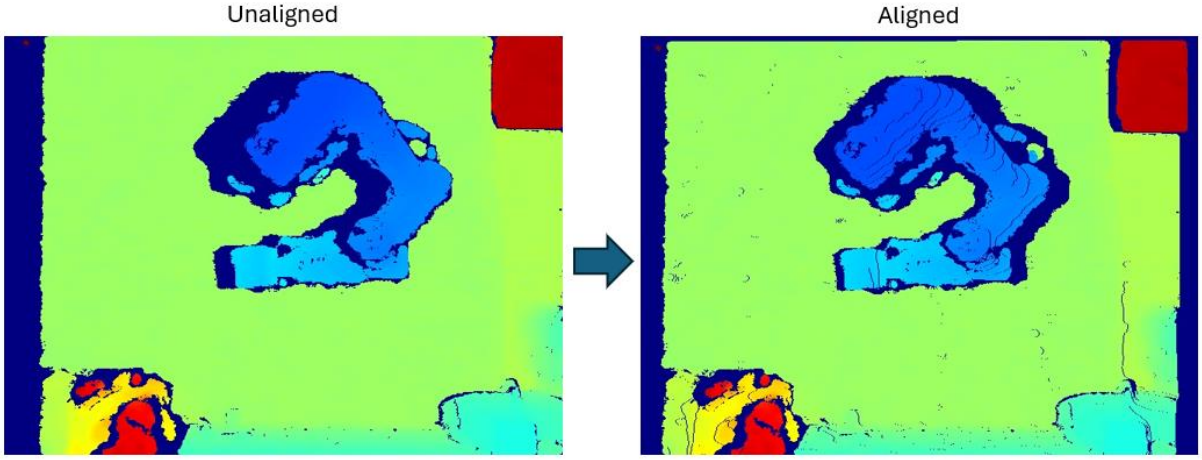


Image 35: Misaligned vs aligned depth

The acquired and pre-processed image data is then passed to specific processing modules based on the robot's commands. These commands are indicated by flags within a shared communication object between the main program and the communication thread. Furthermore, all these flags are managed under a lock mechanism to ensure thread safety. For instance, if the robot requires object detection, the "identify_objects_NN" function is called to locate objects within the workspace using neural network techniques. Alternatively, if the robot needs to determine the finger's orientation, the "Identify_holes" function is executed to detect the orientation of the two holes on the finger object. The detected orientation data is then stored in the communication object for the server to transmit back to the robot. After the requested processing is complete, the corresponding flag is reset, and a signal is sent to the robot, confirming that processing has concluded and results are available.

The main loop continues to await commands and process frames until a shutdown signal is received from the communication thread. When this happens, the communication threading and the camera pipeline are stopped and all the resources are released, ensuring a clean system shutdown.

7.2 Hole detection

The hole detection module is a specialized component of our image processing pipeline, which is designed to determine the precise orientation of the finger object. Upon receiving a command from the robot to capture an image, this module analyses the specific area where the finger is going to be placed to accurately identify and determine the location of its two distinct holes. This information is critical for guiding the robot's subsequent manipulation tasks, as it determines the correct orientation at which the finger must be placed on the mounting process.

The hole detection process is executed by a function called "Identify_holes", which is called from the main program. This function orchestrates a series of image processing steps to pinpoint the two holes on the "finger" object and determine its orientation.

As the main program provides the function only with the general RGB image, the first step involves cropping that image to the exact ROI. This cropped area corresponds to the known, constant location where the "finger" object will be placed each run, with a very small slack to accommodate minor positioning variations. Fixing this region as tight as possible is particularly important, as it makes the code more stable and reduces noise detection from possible objects located at the background quite considerably. This way undesired noise posterior errors are avoided, and the model's correct functioning is ensured.

The cropped image is then processed to identify the object's boundaries. These boundaries are detected using a semi-fixed manual segmentation approach. This approach is different to the one used for object identification that will be later explained and resembles more to the old programs that used manual segmentation instead of relying on NNs. The method consists of using carefully selected thresholds adapted to changing lighting conditions, which are dynamically determined by a "threshold.json" file. This file is generated by an auxiliary calibration routine that enables a manual threshold selection by the user at the beginning of each run, which reads some user-friendly interactive windows shown on the Image 36. When this detection is done, an object-mask is created, where just the localized object will appear on the frame. Therefore, distinguishing the object from its background.

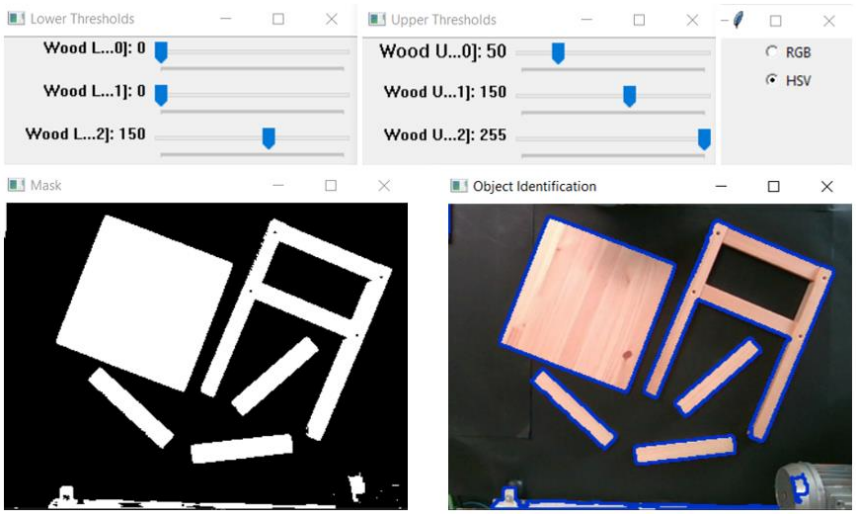


Image 36: Lightning-condition based threshold adjustment

Afterwards, this new mask is passed to another function called “detect_holes_in_object”. This function is responsible of detecting the holes inside this mask. To detect those holes, the green channel of the mask is first isolated. This is because as seen on the Image 37, it is the one that provides strongest contrast between hole and object no matter the lightning conditions. An Otsu's thresholding method is then applied to segment the image, distinguishing the holes, followed by an inversion to ensure the potential hole regions are represented as foreground pixels.

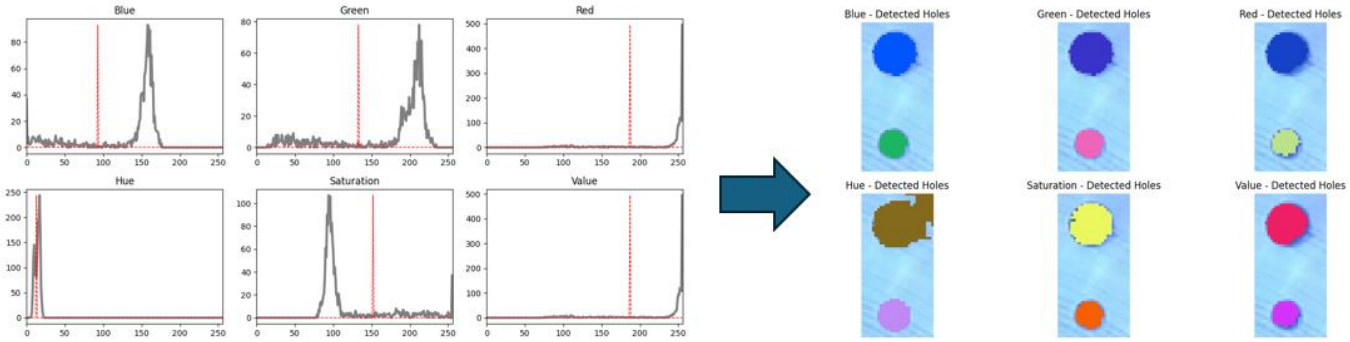


Image 37: Hole detection mask comparison

Following this, morphological operations are performed. Specifically, a morphological closing operation is applied. This operation, which consists of a dilation followed by an erosion, is crucial for filling small gaps or breaks within the hole contours, making them more homogeneous and robust for subsequent detection. This transformation seen on the Image 38 ensures that a single continuous contour is detected for each hole, even if there are minor imperfections in the initial segmentation. Subsequently, the system searches for external contours within the morphologically processed image, with each detected contour representing a potential hole boundary.

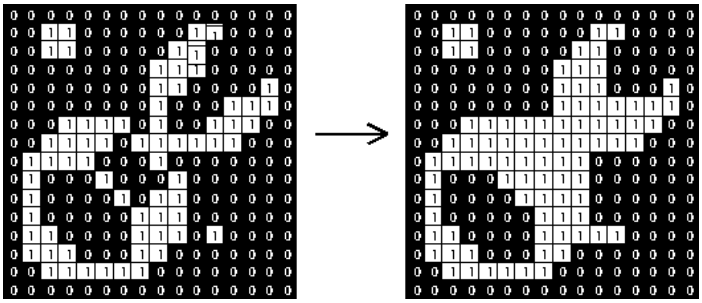


Image 38: Closing operation *[34]*

Each detected contour is then rigorously filtered based on its circularity and size. Circularity, a key shape descriptor, quantifies how closely a contour resembles a perfect circle and is calculated using the formula:

$$C = \frac{4 \cdot \pi \cdot A}{P^2} \quad \text{e)}$$

Where A is the area of the circle, P is the perimeter and C is the circularity. For a perfect circle, C=1.

In this case, a circularity threshold equal to *0.7* is set, a value specifically chosen to effectively filter out non-circular shapes while retaining the highly circular features of the holes. This means that only the contours exceeding this threshold are considered as holes further on. Additionally, a minimum hole radius filter is applied to discard exceedingly small contours. This second filter prevents minor pixel clusters or noise from being misidentified as holes.

Afterwards, for each valid contour, a minimum enclosing circle is fitted and its radius in pixels is converted to centimetres using a predefined pixel to cm ratio. This ratio is fixed because the "finger" object is consistently placed at the same position from the camera. This approach eliminates the need for more computationally intensive methos such as 3D deprojection to determine real-world dimensions, offering a faster and more efficient solution for this specific application.

Finally, a verification step ensures that exactly two holes are detected. If more or fewer holes are found, an error is logged, and the process is typically restarted, as this indicates an issue with the image acquisition or processing. If two holes are successfully identified, they are sorted by their radius in descending order. The orientation determination of the larger hole is then based solely on its y-coordinate relative to the smaller hole. This simplification is valid because the "finger" object is consistently aligned with the y-axis of the camera, making a straightforward vertical comparison sufficient for determining its orientation. This result is the most important one, as the placement of the bigger hole (Upper or Lower) is the one determining if the finger should be rotated one way or the other, therefore being the crucial information the robot demands.

But apart from that, the holes are also depicted on the RGB image for the user to have the ability to check if the detection has been correctly done or not. So, the detected holes are annotated on the original frame with circles and their calculated radii is also depicted for visual confirmation as seen on the Image 39. The “Identify_holes” function then returns the annotated image, the list of detected holes (centres and radii), and the identified position of the larger hole, providing the critical orientation information to the main program for robotic control.

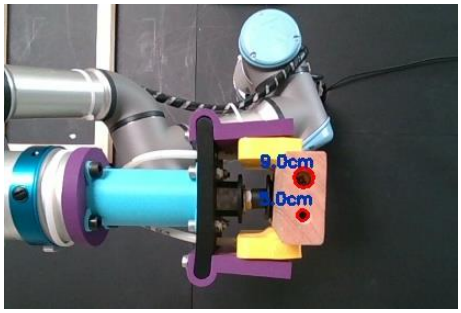


Image 39: Identified holes

7.3 Object detection

The object detection module provides crucial environmental awareness for the robotic system. Its purpose is to identify and localize the objects situated within the designated placement area. This involves analysing the captured imagery to determine which objects are present and extracting essential information, such as their type and precise coordinates, for the robot to effectively interact with its environment. At the beginning, a classical segmentation approach was used to identify the objects. Buth this approach was only able to detect planar objects and could not distinguish superposed objects. Therefore, the initial approach was discarded, and a new method was developed. Nonetheless, the functioning of this old method and all the codes developed for its functioning are more deeply explained on the Annex B.

So as mentioned, everything done until now worked perfectly for clearly distinct objects that are separated from each other. But this is not always the case. So, in order to move a step forward on the project, the ideal case must be left aside, and the program must be adjusted so that it detects 100% randomly placed objects. This means that the objects will also be placed one above the other sometimes and therefore the previously developed segmentation method is no longer valid as the superposed objects have no distinct boundaries with between each other.

To do so, an option would be to work with point clouds, by recreating the 3D space of the environment, as on the Image 40. Nevertheless, the inaccuracy of the camera when doing so is so high that even after applying the best pre-processing possible, it is not possible to achieve a clear enough distinction of the objects to later obtain the desired characteristics.

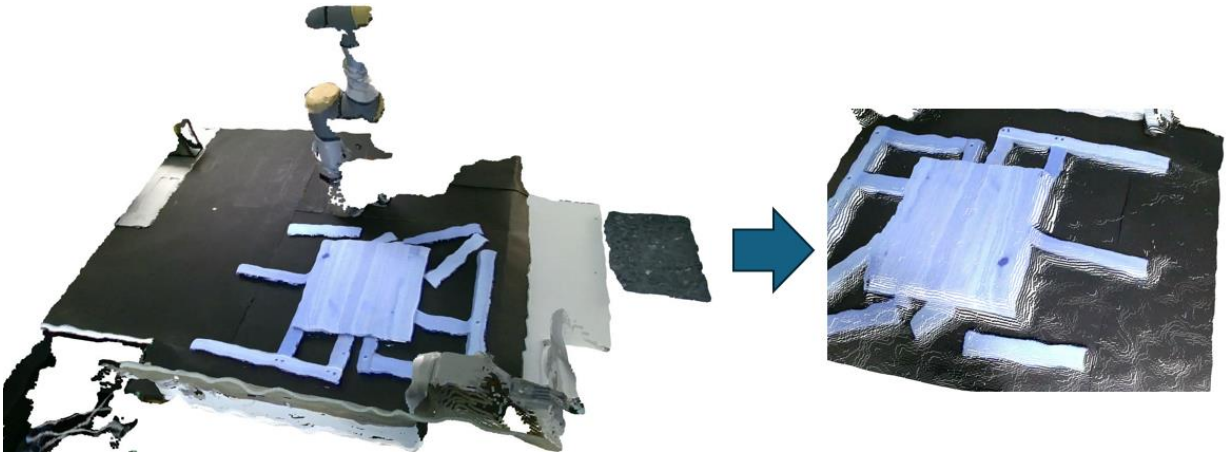


Image 40: 3D point cloud of workspace

Due to this reason, it has been decided to use a completely different strategy: using Neural Networks (NN). Artificial Intelligence (AI) is currently at a developed stage where there are enough tools and standardized processes to create your own model in a quite straightforward way. Yet, this approach could be done in two separate ways: by means of a 3D point cloud based NN, or by means of a 2D image based NN to which you can later add the depth information.

The clear option to select was to develop the 3D NN. This option seamed the most straightforward and complete one, as it can be seen on Image 41, there are more aspects to analyse like three-dimensional spatial relations between points and intrinsic parameters inherent to the camera´s perspective. So, while adding depth information to 2D images significantly improves the capabilities of neural networks for 3D vision tasks, working directly with point clouds allows neural networks to learn richer and more robust representations of 3D geometries. This leads to better understanding of the environment and therefore a better performance in object identification.

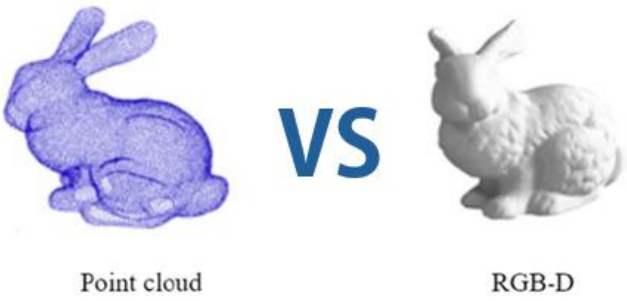


Image 41: 2D vs 3D data *[35]*

However, some incompatibilities between the computer and essential libraries for the development of this neural network made this option impossible. Therefore, the second-best option was selected, developing a 2D neural network. But developing a complete model on your own does not make sense, as there are already really good models that can be imported directly to the project and just adapt it to the specific application. Out of all the possible models "YOLO" has been selected as the best option. There are also some other possibilities such as "SSD" or "DETR", but YOLO offers a faster performance and continuous improvement updates by the developer of the model (*Ultralytics*). This means that if in the short future the project needs to be used, the model could be easily changed for a newer version and its performance or time would increase. To see how fast this could happen, in just 1 year they went from YOLO8 to YOLO11, increasing the performance of the algorithm considerably as it can be seen on the Image 42. More notably, at the moment YOLO11 is the latest version, but they are currently developing YOLO12, which is currently at testing stage.

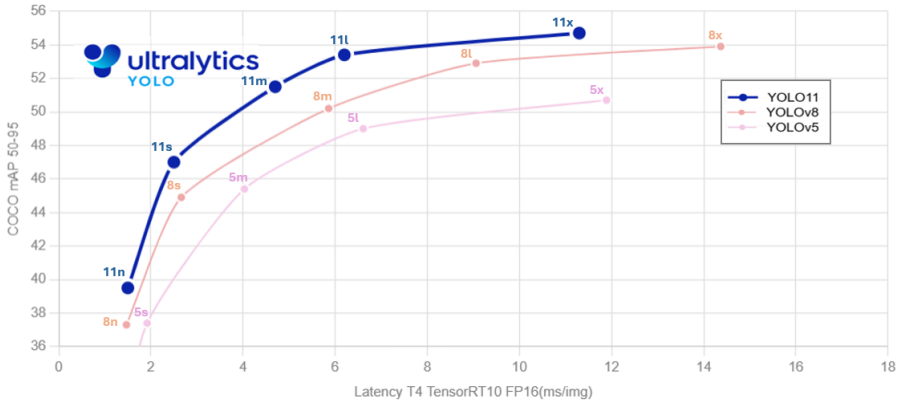


Image 42: YOLO version comparison

On that account, the job of this NN is to identify the upper objects from the camera image so that they could be later treated the same way it was done before applying the NN. To fulfil this purpose, the NN must be trained to identify just the objects that appear complete and have no hidden parts. This means that if a finger is placed above a base, the base must not be detected while the finger should. This method follows 2 arguments. Firstly, by doing this the NN will have less object variability, so it will achieve a better performance for the same training. Secondly, the robot needs just the information of the upper objects as they are the only ones it can pick up without removing any other object.

Nonetheless, this object identification can be performed in various ways, as the YOLO model offers various identification methods as seen on the Image 43. But out of all 4 options, only the "Object detection" and "Instance segmentation" are interesting for this project, as more than one objects will appear on the same image (so we discard "Image recognition") and the different objects must be clearly differentiated between each other even if they are from the same class (so we discard "Semantic segmentation"). As it is difficult to determine which of both remaining options will have a better performance on this project, it has been decided to develop both NN and later compare the results to see which is the best options.

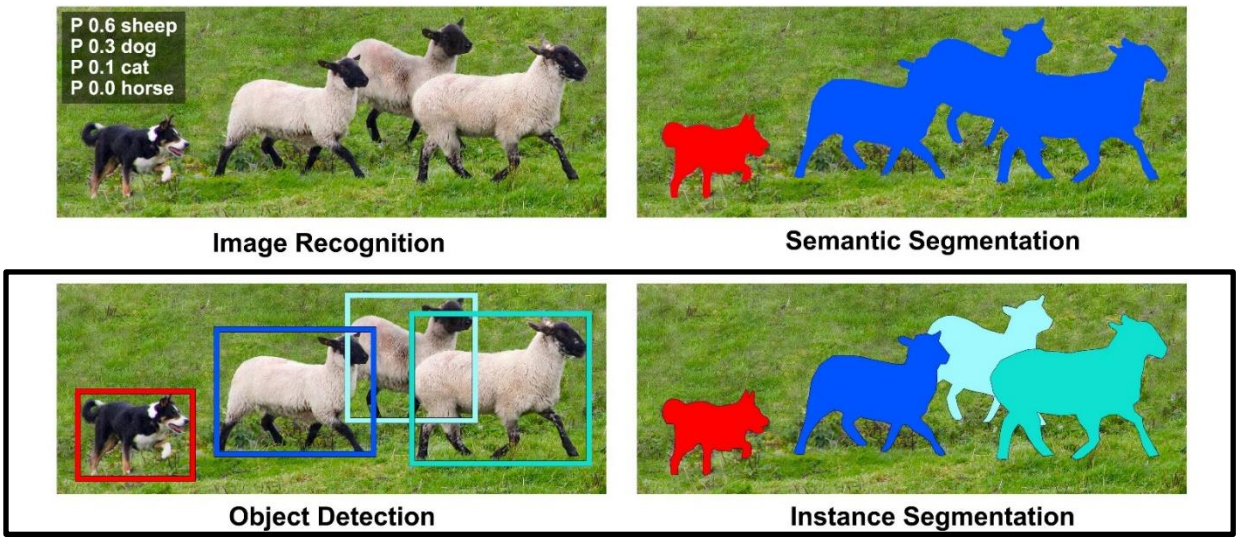


Image 43: Object identification methods

7.3.1 NN training process

Even if both NNs detect the objects in different ways, they both follow the same training procedure with just some minor changes between them. Therefore, their training and validation processes can be explained together and easily compared to each other.

The first step on the process is the Data acquisition step. This consist of taking the maximum amount of picture with the maximum variability possible, so that the NN has more data to learn from and a higher adaptability towards different scenarios. The quality of the images is one of the most important things to take into account in this step. The quality of the images fed into the system has a direct impact on how well it can "see" and understand the environment. If the images are blurry, poorly lit, or just do not show what is needed to see clearly, the system will struggle to identify objects, detect patterns, or make accurate predictions. Consequently, this step is one of the most important in the entire process, as a poor image at the start can have a domino effect in later steps, causing problems all the way down the line.

Knowing this, an effective way to create a solid base with good images is not only to take the maximum number of pictures with the highest variability possible. Other aspects like sensor quality, lightning conditions, an adjusted focus and image stabilization also play a vital role in the quality of the image. Unfortunately, there is nothing to do about sensor quality and being able to enforce adverse lightning conditions is an intrinsic goal of the project, so the other aspects like ensuring a proper camera calibration become vital. Moreover, using some other techniques like Data augmentation also helps increase the size and variability of your dataset. So, by applying some minor changes like rotations, changes in brightness and flipping the images, helps the system become more robust and less sensitive to minor variations in how objects appear.

But the dataset does not only consist of images. So, before applying this data augmentation, the dataset must be fully completed and therefore the images should be adapted to the conditions and correctly labelled. Otherwise much more images would have to be labelled, and the work would be multiplied.

So first of all, the images have been chopped to the object placement area. This means that the images used to train the NN model exclusively show the zone where the objects can be placed, reducing external factors that could act as noise and also reducing the number of pixels to increment the training and image processing speed.

Once this is done, the images have been manually labelled using *Roboflow*, which allows both segmenting the objects and adding detection bounding boxes to them in a really easy and intuitive way. This is a manual time-consuming process, but the tools provided by *Roboflow* make it faster and also generates the dataset automatically, so it is ready to download and directly use in the code. Here is also important to mention that the objects have been labelled following the previously explained criteria. This means, just labelling the upper objects and generating to separated datasets with 1 type of label each: in the first one the bounding box of the object needs to be specified in order to train the “Object detection NN”, and in the second one the contours of the objects are the ones that have to be labelled for the “Instance segmentation NN”.

In the aftermath of the labelling, all the images were also resized from a 480x640 frame to a 640x640 frame. 640x640 pixels is the recommended input size for training and inference in YOLO, as its architecture is designed and optimized to work efficiently with this input dimension. Also, it can be seen on the Image 44, that the same dataset with a 640x640 image size performs better than the one with a 480x640 sized images.

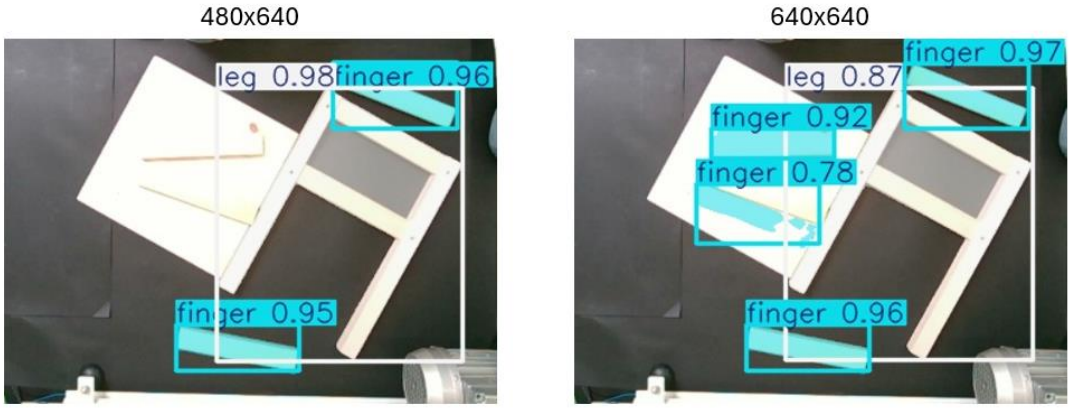


Image 44: 480x640 VS 640x640 image size performance

After completing this, the dataset is almost ready, just having to separate the images in the different batches: training, testing and validation. For this project there is no specific requirement with respect to the image classification. Therefore, standard data classification measure of *70% training + 20% validation + 10% testing* has been used, separating the images randomly into each of the batches.

Finally, with the dataset completely ready the NN training starts. Here, based on a *COCO* pretrained model, the NN starts to adjust his internal weights to the provided dataset for the amount of stablished epochs. The most important aspects to consider on this step are the batch size and number of epochs used. The number of epochs will determine the amount of iterations which the NN goes out through. This means that if the model uses 20 epochs, it will perform 20 backward and forward passes, each time readapting the weights to approximate convergence to the local minima of the system. On the other hand, the batch size is the number of training examples processed before the model's weights are updated. This means that for a batch size of 3, the model will make guesses for all these 3 images and then look at all their answers together to give a jointly feedback on how to adjust the weights.

Here it is important to mention the importance of finding the correct number for these 2 parameters. At first sight it might seem that selecting a small batch size would be the best option, but that is not true. This would have a slow convergence, and the loss functions could fluctuate wildly between iterations, making it harder to monitor convergence. But even if bigger batch sizes offer faster training time and more stable gradients, they also have some issues like the risk of getting stuck on a sharp minima. So, whereas smaller batch sizes might be beneficial to provide more frequent updates and a potentially better generalization, larger ones can be more efficient and still provide a reasonable gradient estimate. *[36]* Having all this into account, for the project the values of *batch size=16* and *epoch=100* were selected, allowing to have an intermediate batch size that does not get stuck as easily on a local minima while also ensuring a more stable gradient and a faster training time. The high number of epochs also helps the model explore the loss landscape more thoroughly, giving it more opportunities to potentially escape these plateaus and find a

better minimum, also acknowledging that a higher number of epochs will only increase the computational time and not improve the performance. These values have been obtained throughout gradually fine tuning the model until the obtention of a balanced relation between the parameters that create a good model estimation environment.

On each run, for adjusting the weights of the model to the correct values, the NN performs the following process in order, following a scheme similar to the one on Image 45:

- **Forward Propagation:**
  A batch of training data from the training set is fed into the input layer of the network. This inputted data flows through the network, layer by layer. At each layer, the data is transformed by the weights. Finally, the network produces an output prediction for each example in the batch.
- **Loss Calculation:**
  The network's predictions are compared to the actual labels corresponding to the input batch (ground truth). From here, a loss function quantifies the difference between the predictions and the true labels. This loss value indicates the network's prediction accuracy for the batch.
- **Backward Propagation:**
  The error is propagated backwards through the network and the gradients of the loss with respect to each of the network's weights are calculated. These gradients indicate how much each weight contributed to the overall loss.
- **Weight Update:**
  An optimizer algorithm uses the calculated gradients to update the network's weights. The goal of the update is to adjust the weights in a direction that will reduce the loss in the next forward pass, leading to better predictions over time.

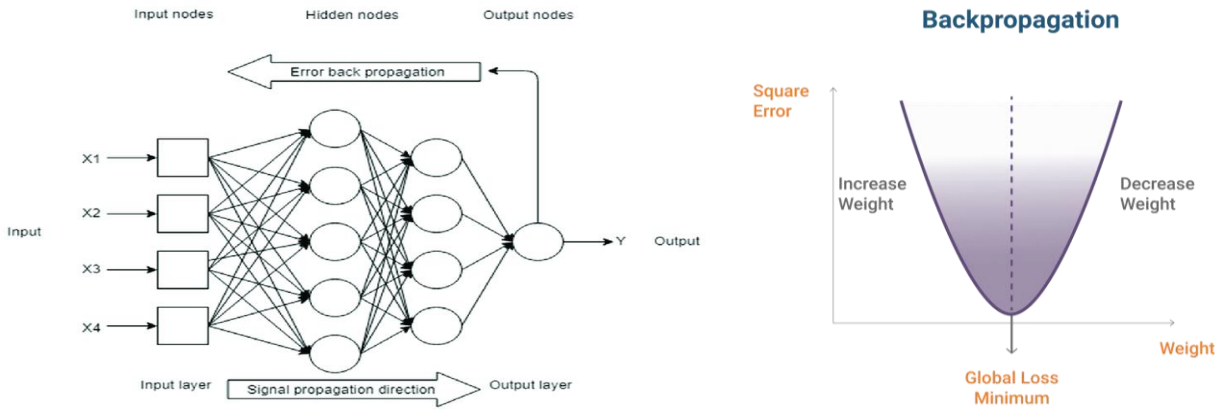


Image 45: NN training process (propagations) *[37,38]*

At the end, for a final check of the obtained model performance, the model is subjected to the testing batch of the dataset, as this is a previously unseen data. As, the testing dataset is fed into the trained neural network, the network produces predictions for each example in the testing set. The predictions are compared to the true labels of the testing dataset, and depending on the obtained values, the performance and errors of the model are measured. If these labels match the ones from the dataset (the correct ones), the precision of the model would be of a 100%. But before checking any result, is important to understand that the measured errors can be classified in three types:

- **Box Loss:**
  Quantifies how well the predicted bounding boxes around objects match the ground truth bounding boxes. It penalizes inaccuracies in the predicted box's position and size.
- **Seg Loss:**
  Evaluates the accuracy of the predicted masks for each detected object. It compares the predicted segmentation mask (which pixels belong to the object) with the ground truth mask.
- **Class Loss:**
  Evaluates the accuracy of the predicted class label for each detected object. It penalizes incorrect classifications of the objects.

7.3.2 NN result comparison and selection

Overall, by analysing those results along others such as the confusion matrices, all the models can compare between them to select and use the best one for the project. Nevertheless, in order to better understand the results, they will be analysed one by one, explaining each concept and comparing both the. And as in this case the two models to compare are the "Object detection model" and the "Instance segmentation model", each of the analysed parameters will contain both respective results.

The confusion matrices can be the first parameter to compare, as they may be the most representative and intuitive indicators of the model's accuracy. A confusion matrix is a table that summarizes the performance of a classification model, like a neural network, on a set of test data. It essentially measures which classes are predicted correctly versus incorrectly. Here, the diagonal elements show correctly predicted elements, while the off-diagonal elements show the different types of mistakes made by the model.

Moreover, this confusion matrices can be normalized to ensure a better comparison, as it gives you an immediate sense of the proportion between errors and correct predictions. If the matrix is normalized by column (actual class), the recall for each class can be easily seen. If it is normalized by row (predicted class), you can see the precision. Therefore, normalizing the entire matrix gives you the overall proportion of each outcome. By doing so, the results sawn on the Image 46 are obtained. *[39]*

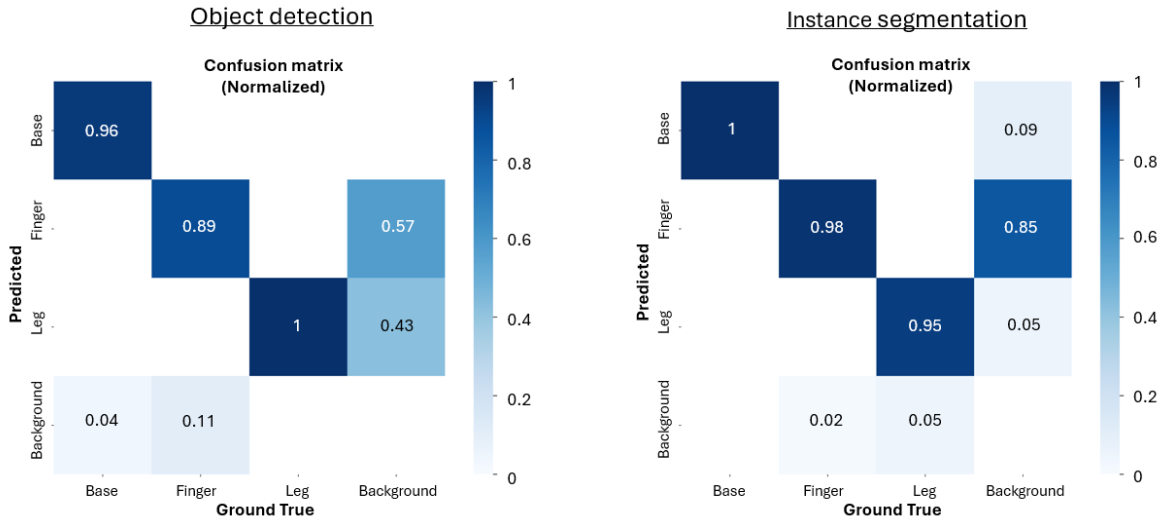


Image 46: Confusion matrixes (normalized)

Whenever both confusion matrices from Image 46 are compared, it can be seen that even if both models perform correctly, the instance segmentation's accuracy is almost 3% higher for the objects. Even the mispredictions of the background are much more concentrated on "Finger" labelled predictions, which makes more sense, as a small outstanding part of a leg placed underneath other objects could easily be mistaken for a finger due to their resemblance. Nevertheless, this increase in precision is not unexpected. Because, while both models work by identifying and locating objects, Instance segmentation provides a much richer understanding of the scene, which is reflected in its confusion matrix.

For a deeper understanding of the results, the confusion matrix itself has been used as the basis for calculating many other important performance metrics. But for this, the non-normalized confusion matrices must be used: *[40]*

- **Accuracy:** Ratio of number of correctly classified instances to the total number of instances:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad \text{f)}$$

  Where, TP is true positive, FP is false positive, TN is true negative, and FN is false negative.
- **False positive rate (FPR):** Rate of wrongly classified instances (Low FPR is desired):

$$FPR = \frac{FP}{FP + TN} \quad \text{g)}$$

- **Sensitivity:** Proportion of positives that are correctly identified:

$$Sensitivity = \frac{TP}{TP + FN} \quad \text{h)}$$

- **Precision:** Ratio of positively predicted instances among the retrieved instances:

$$Precision = \frac{TP}{TP + FP} \quad \text{i)}$$

- **Specificity (SP) / True negative rate (TNR):** Proportion of negatives that are correctly identified:

$$Specificity = \frac{TN}{FP + TN} \quad \text{j)}$$

- **Recall:** Ratio of positively predicted instances among all the instances:

$$Recall = \frac{TP}{TP + FP} \quad \text{k)}$$

All these parameters have been calculated for all the labels of each of the models and depicted on the following Table 12 for a clear comparison between models.

Table 12: Performance metrics of the NNs

	Base		Finger		Leg	
	Obj. identify.	Instance seg.	Obj. identify.	Instance seg.	Obj. identify.	Instance seg.
Accuracy	99,13%	99,31%	91,41%	90,51%	97,71%	97,88%
FPR	0,0102	0,0000	0,1422	0,0752	0,0274	0,0073
Sensitivity	100,00%	96,43%	89,15%	98,33%	94,84%	100,00%
Precision	94,49%	100,00%	84,89%	94,46%	87,80%	98,35%
SP / TNR	98,98%	100,00%	85,78%	92,48%	97,26%	99,27%
Recall	94,49%	100,00%	84,89%	94,46%	87,80%	98,35%

As with the confusion matrix, this Table 12 also shows the better performance of the Instance segmentation. So, even if not all metrics, the majority of them show that the Instance segmentation is a better model for the project. The only two exceptions out of all the metrics were the base sensitivity and the finger accuracy. This might seem like a weakness in the model, but in reality, is a good result for the 89% of the metrics to agree with each other about which is the best model.

Besides, those possible weaknesses can later be solved by adding new info to the dataset and improving the model's robustness. In this case, more images with base and finger labels would have to be added. But this might also not be necessary as those values are not necessarily bad, only a little worse on a comparison with another model. In reality, all of the metrics are higher than 90%, which is a decent result. And even if they could be improved, that would have to be determined by other factors, as there is not any visible reason to do so just by observing these metrics.

In addition to confusion matrices and their derived metrics, there are several other critical aspects of a neural network's performance that can be analysed and compared. For example, the 3 types of Losses that have previously been mentioned and their mean average precision (mAP).

The different Losses have already been defined, but the in order to analyse the mAP as well, it is important to understand what it represents. mAP is a widely used metric for evaluating the performance of object detection and instance segmentation models. In essence, mAP tells with a percentage, how precise a model's detections are, considering how many actual objects it found and how well it localized them. But there is not a unique type of mAP. These types of models can be measured using 2 types of mAP:

- **mAP@0.5:** Shows if the object and its class are being correctly predicted. It does not consider localization and shape.
- **mAP@0.5:0.95:** Apart from the previous, it also takes into account if the exact location/shape of the objects are being correctly predicted. (More demanding)

Both types of mAP as well as the Losses are displayed on the graphic sawn on Image 47 for both models, creating an easy visual comparison of their evolution through time while still providing the final value of each metric at the end of the training.

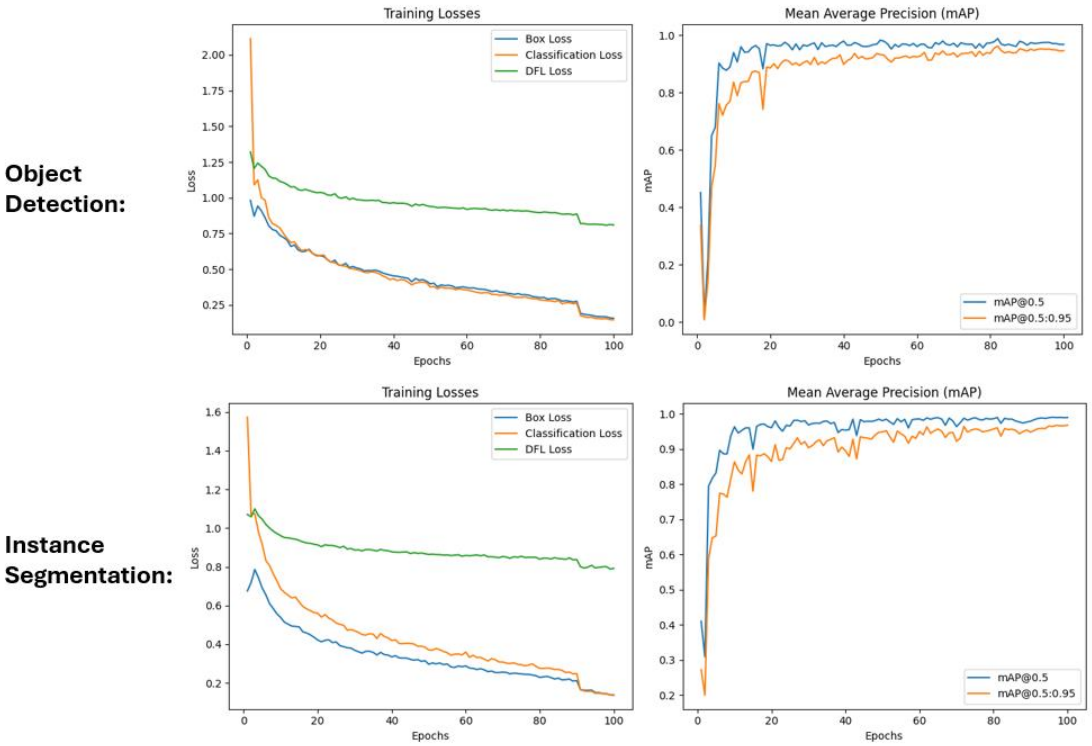


Image 47: Loss and mAP evolution comparison

As the shown on the graphics, even if they have a different evolution through the epochs, both models end up obtaining really satisfactory results where there is not such a noticeable difference between them. For example, by analysing the best value of the mAP@0.5, the object detection model obtains a *98.8%* accuracy while the instance segmentation model obtains a *98.98%* accuracy. And the same thing happens when analysing their mAP@0.5:0.95, where the object detection obtains a *96.2%* accuracy whereas the instance segmentation has a *96.77%*. In both cases the difference does not reach even the *1%*.

The same thing happens when comparing the Losses. The Object detection model has a Box Loss of *0.2868* and a Class Loss of *0.3227*, while the Instance segmentation model has a Box Loss of *0.2174* and a Class Loss of *0.2278*. In both cases the difference is less than *0.1*, which is not that big. So, even if the Instance segmentation model shows a slightly better performance in mAPs and Losses, the difference is not big enough to notice a significant performance improvement with respect to the other.

But after all, the most intuitive way to compare the performance of these two types of models is simply by observing the output responses. Sometimes all the parameters or metrics can point one way, but if the visual results point another way it might be a better indicator. So, even if it is especially important to analyse all the previous parameters the visual validation cannot be forgotten.

So, if both models are run on the same set of test images as on Image 48, the performance of the models can be directly checked looking at various factors such as the following: better bounding box / contour fitting, higher confidence, less detection errors such as false positives or false negatives, better handling of occlusion…

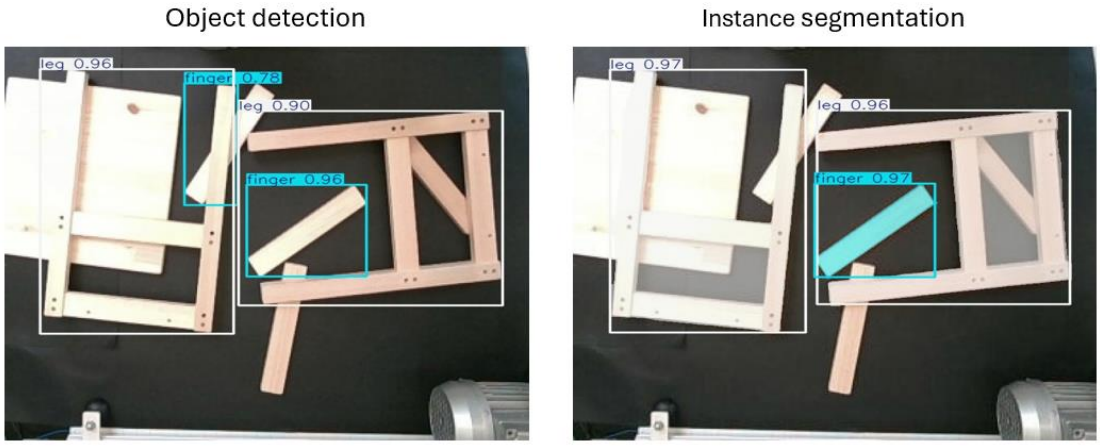


Image 48: Comparison of visual responses between NN models

All in all, and reassuring the results obtained from the metrics comparison, this image shows how the Instance segmentation model handles the situation better and offers a more robust response. For example, the first image (Object classification) detects a false finger from a leg part and a semi-occluded finger while the second image (Instance segmentation) does not. Moreover, all the detected images also show a higher confidence and a better adjustment of the bounding boxes, which is a signal of a better performance. It is also important to notice that this Image 48 is just an example of the batch that represents the difference between both models, but this differences along with others are noticeable throughout most of the images in the batch.

7.3.3 NN validation

Once having selected the appropriate model, the last step is to reinforce the model and validate its performance. This means analysing the lacks of the model and trying to fill the gaps so that it can perform correctly against any circumstance.

The first step to do so is to add more images to the training. But not any image is necessary. For example, the image is robust enough against all lightning conditions, so the brightness of the new images to add does not matter. On the other hand, it has been observed that sometime the object predictions are not accurate enough, so this must be reinforced adding more images with those unnoticed or mislabelled objects. As the confusion matrix shows, the base objects are much better detected than the legs or the fingers. This means that more images containing those two objects must be added to the training dataset. Additionally, from other images that were poorly predicted must also be added to the training. Those can either be new images obtained from the streaming or images directly obtained from the testing or validation datasets.

Once with a robust training dataset, the model has been retrained. But this time more epochs were used to obtain a more robust result. Even if it is not necessary and the obtained improvements are not significant enough for the computational cost, *300* epochs were used instead of a *100* for this model. This has been done because for the final model every small improvement counts it is worthy although the high computational costs at time consumed. At the end, the model obtains the following results shown on the Image 49.

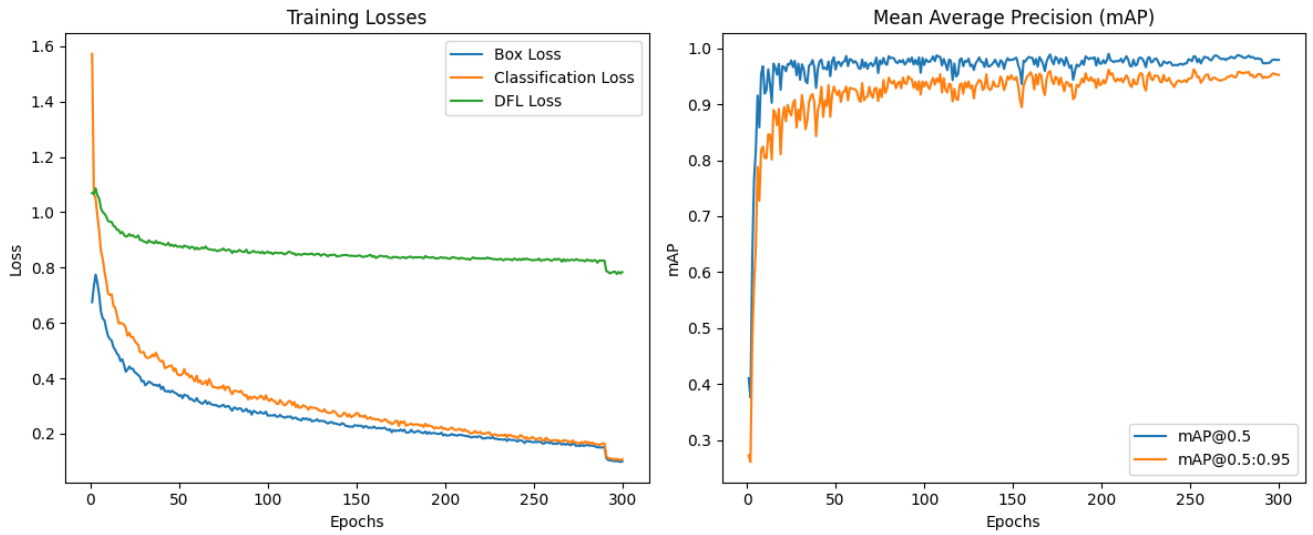


Image 49: Loss and mAP of final model

The final results of the graph being the following on Table 13.

Table 13: Final model's performance metrics

mAP@0.5	mAP@0.5:0.95	Box Loss	Class Loss	DFL Loss
98.61%	96.24%	0.2218	0.2726	0.8247

The metric results might seem worse than the previous model trained with *100* epochs, but as earlier mentioned, this is not the most accurate way to see a model's performance. This decrease on the metrics can just be a consequence of having a bigger and more complex training dataset, and even in that scenario the metrics are still really similar, and it can be said that their decrease is insignificant. So, the real performance increase might be confirmed by looking at the visual results obtained. Here is where the real improvements are shown. Now the model is more likely to make a good prediction and even if the improvements are not that high, the model now make less mistakes than previously and is ready to be used.

Finally, once the model is correctly trained and reinforced, it must be tested and implemented on the real application. This means that the model is downloaded from the code used for model training and then uploaded again on the main code where the camera images will be analysed on the real application of the project when the robot demands it. It is an easy step to follow, but vital to check the correct functioning on the model on real applications, ensuring the compatibility between procedures and that all the previous job has being correctly done.

7.3.4 Implementation of the NN on the code

The trained YOLO model is implemented on the code as a function. This means that the main code only needs to call the function each time a new frame is captured, leaving a clean code that is easy to understand as it only needs one line. Here, by passing some minimum information such as the RGB and Depth images captured the function performs the predictions internally and returns all the predictions as well as the predicted images to the code, so that they can be used and displayed.

This function will also replace the previous object detection functions, mentioned on Annex B, as they are no longer valid and cannot detect superposed objects correctly. This means that the code used to adapt the threshold to the changing lighting conditions is also not necessary, as it was used to regulate the detection masks of the mentioned object detection function. This is due to the NN's advanced ability to detect the objects at any lightning conditions and even if they are superposed with each other, which is a significant step forward not only because it will perform better but also because less functions are being used to do the same job, which is much more efficient and keeps the code more understandable. Nevertheless, these unused codes will not be erased just in case they are needed in a near future.

For a better understanding of the new object identification function based on the YOLO NN, it is necessary to know how the information given by the main code is processed to obtain the desired output information. So, as previously mentioned this function receive the input images from the camera and also its intrinsic parameters to later give 2 outputs: the predictions and the RGB image with the predictions and some other information displayed on top. But apart from that, it also obtains some necessary information from those predictions to create a .JSON file that can later be transferred to the robot with all the information it needs. This file contains not only the predicted label of each object, but also its confidence, rotation angle, size and holding position where the robot needs to go. All this necessary information is extracted from the predictions, whether directly or indirectly in several ways.

For example, the labels and the confidence of each of the detected objects are obtained directly, as the prediction already provides the class name to which each object belongs along with the confidence of the object's detection. So, these characteristics are directly stored on the .JSON file for the robot to process them. Elseway, the confidence value is used to order the detected objects from highest to lowest confidence. By doing so, it is ensured that the robot first looks for the safest predictions. So, if the camera has detected some noise or has mis-predicted an object, it will have a lower confidence percentage, being highly possible that on the next image (when the robot has already picked up one object) the scene will be clearer and thus the object will be better predicted. This helps not only increasing the model's safety, but also its performance accuracy.

But on the other hand, there are parts of the information that need to be processed in order to obtain the desired information, as apart from the previously mentioned the predictions only provide with the bounding box and the detected contours of each object. From here for example the rotated bounding box can be obtained, that can be used to determine the rotation angle in which the object has been placed. This information is really useful for the robot, as the tool needs to match the object's angle in order to pick it up correctly. Otherwise, the robot's tool would collide against the object and an alarm will stop the process, which is highly undesired.

The same rotated bounding box can be used to measure the object's size. To do so, the 4 corners of the object are deprojected to the 3D coordinate system using information from the depth image and the distance between corners is then measured in centimetres. Also, to ensure this measurement is more robust and as each side is measured twice, the means of both widths and heights are calculated. But those width and height values are stored randomly, so afterwards they are sorted so that the longest side is stored as the object's height and the shortest side as the width. But sometime the depth mask can provide inaccurate measurements with noisy data. And if 1 of the 4 corners coincide with one of those noisy points, they will

appear as extremely high or low points on the 3D world. To avoid that noise, a filtering technique has been applied, which detects those outstandingly different points and provides them with a new depth value corresponding with the mean depth of the whole object.

Another characteristic that can be extracted from this contour is the holding position for each object. Apart from the angle at which the robot needs to pick up the object, the exact location where that object needs to be picked up is also really important. Luckily, for the fingers and base object the holding point is exactly the centre of the object, so it can be directly obtained by calculating the centre of the bounding box and deprojecting that point to the 3D coordinate system. But for the leg objects that is not the case. These objects have the holding point on the central wide axis, but on the height axis this position is shifted in order to coincide with the centre of the wooden bracket responsible for maintaining the leg's shape. In order to achieve that, an imaginary line was traced from the centre of the object parallel to the height line. From that central point to the outside each pixel was afterwards inspected one by one on both directions until one of them coincided with the object mask created with the contour boundaries. That point would be the border of the bracket, so as the width of that bracket is fixed it is just necessary to displace the point a couple of centimetres on the same direction to obtain the holding point of the leg. This transformation of the leg's holding point can be seen on the following Image 50. After having deprojected this point as well, all the holding points from all the objects are multiplied by a transformation matrix before being stored, so that their position is updated with respect to the robot's base instead of the camera's.

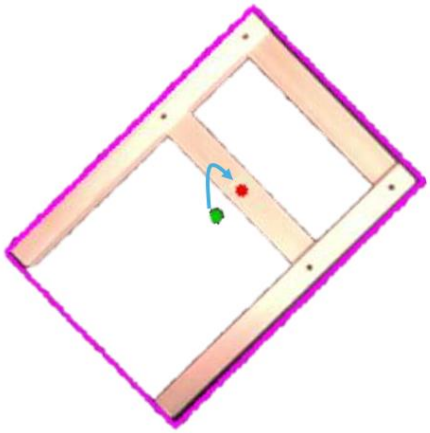


Image 50: Holding point transformation on "Leg" objects

Leaving aside the information extraction methods, this function also has some other functionalities. For example, when the NN was first implemented to the camera it was noticeable that sometimes the contours were not perfectly predicted. They tended to have some small noise points. These were small points placed apart from the actual contour that were considered as part of it. Sometimes they were small points of other nearby objects or just meaningless points. Nevertheless, before treating any contour for their posterior information extraction a noise suppression filter was stablished to erase those noisy points. This is something crucial, as the contours are used for the extraction of most of the major features and a small perturbation in them can cause a big variation on crucial characteristics such as the size of the object or the position of the holding point.

Based on this information obtained from the NN model's predictions, the function stores all the obtained information on the .JSON file and stores it on a folder so that it can be sent to the robot anytime and with the demanded structure. Apart from that, this information is also displayed on an image so that the information can be clearly shown to the user. This display contains the original RGB image provided but with all the detected objects marked and numerated, with some information like the size and angle of the object as displayed on the Image 51.

Image 51: NN based object detection's output image

8. Main problems

During the development of this project, unforeseen challenges appeared. Those problems range from minor inconvenience to more serious problems that incurred delays and greater resource consumption. Although, the smaller ones were easily fixed, causing minimal impact, others required a lot of time and effort, affecting both the timeline and overall development of the project.

Therefore, the identification and analysis of such challenges is crucial for the project, helping to determine the root causes of the problems and estimating their impact on the project. This analysis of the main encountered issues will allow to remark important lessons that will improve planning, decision-making, and risk management in future projects.

8.1 Camera inclination

The first encountered problem has been the inclination of the camera placement itself. Even if the structure where the camera is held is solidly attached to the sealing, this does not necessarily mean that the structure itself is perfectly straight. Even if just one of the parts is attached with a small angle or one of the objects has a small fabrication error, this can be propagated and the impact increased along the structure. So, as in this case the structure is long and consists of various parts attached to each other, it is really normal that its end is somehow tilted.

This apparently minor problem has significant impact on the obtained image, as an apparently levelled floor would look like a slope. This can cause serious problems when detecting object heights, as their value would be different depending on which side of the table they are placed, causing not only disparities between objects but also obtaining different depth results along the same object. If we add the fact that the depth channel of the camera is one of the most critical aspects to take into account in the project, as it has previously been said, it is indispensable to obtain a homogeneous and stable result with respect to this factor. Consequently, the camera needs to be perfectly aligned with the table where the objects are placed, obtaining a reliable result where the table plane seems levelled as it can be seen on the Image 52.

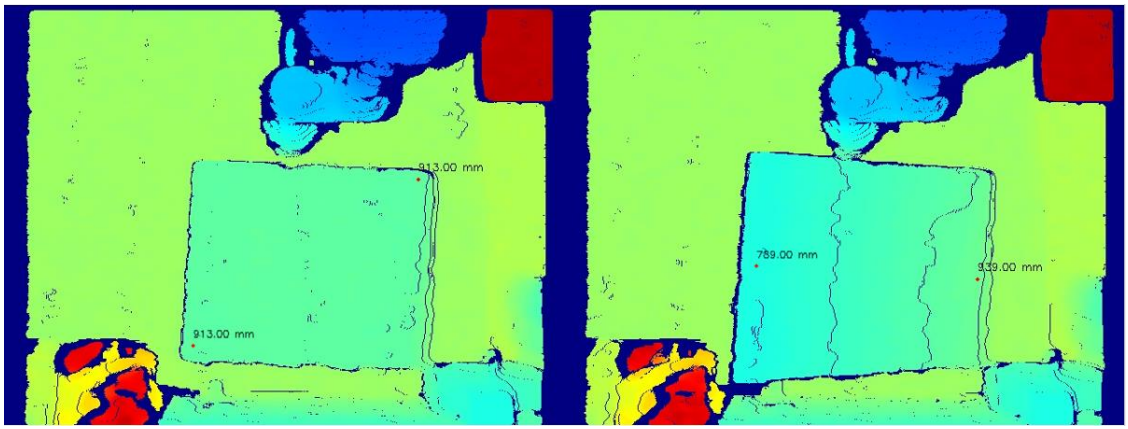


Image 52: Levelled vs. unlevelled surface

Therefore, the structure needed to be completely changed to be more solid and with the fewest number of elements possible. But even if this change was made and the maximum straightness was tried to be given during the mounting, the captured camera image was still not good enough.

So, by acknowledging that problem and concluding that it was almost impossible to mount a perfectly straight structure, it was decided to design an extra levelling tool to achieve this objective. This tool is the one shown on the Image 21 from the Section 4.1 and it basically consists of a self-adjustable structure composed of 4 bolts. So, each of the bolts along with two screws, can adjust the height of one edge of the camera plane. So, by changing these heights one by one and tightening the screws to its place, the plane can be adjusted until the measured depth in all the 4 edges of the image give the same depth value.

8.2 Image misalignment

Another problem it was needed to deal with was the misalignment between the obtained colour and depth frames. So even if both images are captured with the exact same resolution, the pixels of one image did not match the other.

This is a problem created due to the methodology used by the camera to calculate depth. The camera is a stereo vision camera type; this means that the camera captures 2 images from 2 cameras and then measures the difference in pixels between a same point to determine the distance just by knowing the distance between both cameras (baseline). This causes the formation of invalid depth zones were both images do not overlap, as explained in Section 3.3.3.1 (Invalid depth band). This means that that one region is just visible from one of both cameras, and therefore its depth cannot be calculated, obtaining pixels without any depth data.

Those regions cause the image to be extended, and therefore the image to occupy more pixels that do not match at all with the ones from the RGB camera. This means that whenever a pixel from the colour camera needs to be referenced to its analogous in the depth frame, they do not match at all or can even be empty spaces. That being so, a solution needed to be found in order to able to properly match the pixels between both images or their information would never be combined.

On that account, various methods were tried for trying to solve this problem. One of those methods could for example be using the "pixel reference between frames" given by the *RealSense* library. But in any case, even if this method should be quite straight forward and easy to implement, it does not seem to work. Maybe it is a problem of how it was coded, but for all the different methodologies tried no proper results were obtained.

However, the manufacturers know about this problem and provide with a solution integrated on the camera itself. This solution consists of using an alignment function from the same library. This function selects one of the frames as the basis and by using this simple function directly deforms the other frame so that both images match and all the pixels representing one same point coincide in both frames. In this case the depth image was the one decided to deform, as it is the one that comes in a misshapen way from the camera while the RGB frame needs no alteration. The correction performed by this alignment process is shown on the following Image 53, clearly showing the improvement on in between frame pixel correspondence.

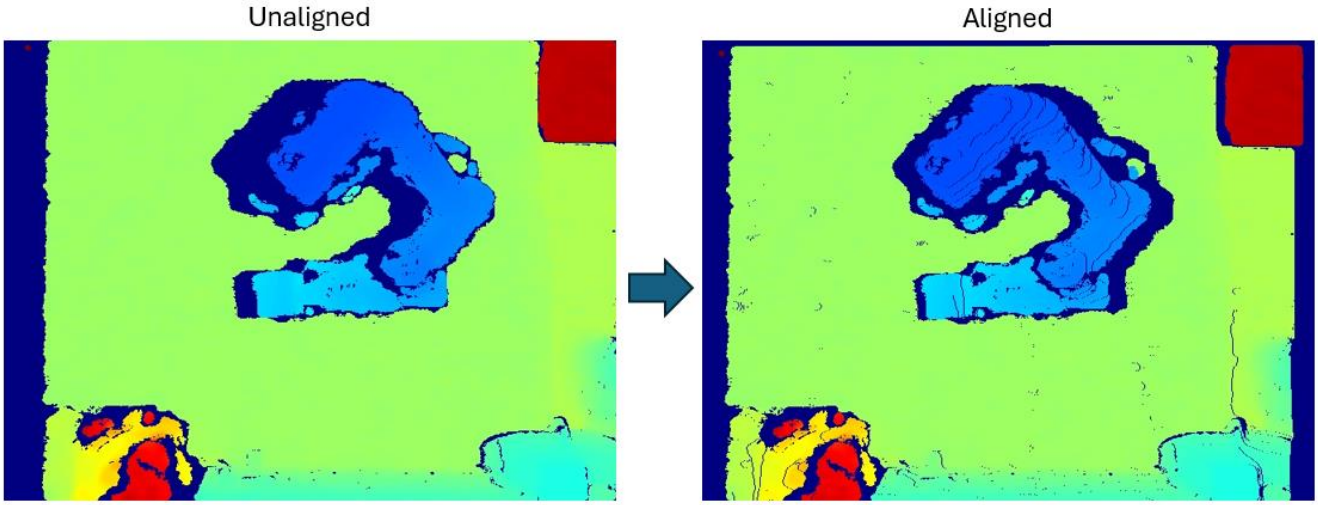


Image 53: Aligned vs unaligned image

8.3 Point obtention

One of the biggest problems for the project has been the point measurement errors generated by the old object detection programs. Due to several reasons such as the mediocre quality of the camera's depth frame, an apparently easy task of pixel treatment and deprojection becomes a problematic and really challenging one.

8.3.1 Empty depths

This program relied strongly on the 3D deprojection of the pixels, and therefore also on the correlation of pixels between RGB and depth frames. But, as it can be seen on either picture of the Image 52, in every point where there is a change in depth some empty valued pixels appear. This "holes" on the depth frame caused a lot of errors when doing the correlation between pixels, mostly on the object edges where there is a higher depth change with their neighbouring background pixels. This was not such a big problem for other measured points, but the edge detection of the objects and their deprojection to the 3D world was an essential part of this code.

Those points were used to measure the width and height of the objects, which was later used for both object classification and angle definition. So, if one of the edges has an incorrect or even empty depth value, its deprojection will be way miscalculated. This would cause the obtention of nonsense width and height values that drag the error throughout the rest of the code, obtaining absolute nonsense values on the information transmitted to the robot.

As a consequence, different approaches were tested for trying to solve this problem. The first approach consisted of detecting those misread depth values and replace them with valid ones. Basically, after measuring the depth value of the 4 edges of each object, the 0 valued depths were replaced with the mean value of the rest of the edges. This approach could be really useful if only 1 edge was misread at a time, but that was not always the case. Sometimes even 3 points were being misread at the time and therefore all the edges will have the same value. The trust on the measurements of the camera must be really high for this

approach to be feasible, but that was not the case as sometimes the points are really diverted from their real location.

As an alternative approach, edge smoothing techniques were also tested. This technique relied on a contour approximation function that served as a noise filtering technique for the detected contours of each object. This function works, erasing the contours that are too close from each other depending on their deviation from the stablished arch-length limit. In another words, the maximum distance from the initial contour to the approximate contour is defined and the contour points outside that limit are erased. As an example of how this function works, on the Image 54 2 approximations for the same object are shown, one with an epsilon of 10% arch-length at the left and another with an epsilon of 1% arch-length on the right. *[41]*

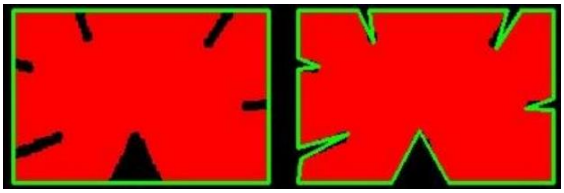


Image 54: Contour approximation technique *[41]*

Seeing that this technique did not work either, a last option was tested. This approach consisted in measuring inner points of the objects instead of focusing on the edges directly. By moving the edges a bit closer to the centre, the depth holes are mainly avoided. So even if the obtained width and height values will be a bit distorted from the real ones, their proportion will still be maintained correctly.

As a final result, a combination of the 3 approaches was used, as it was seen as the best combination to solve the problem. The first one avoided punctual depth misreadings on singular points, the edge smoothing avoided taking measurements on possible noisy contour points and the third one helped avoiding the "depth holes" on the exact edges of the objects. This combination provided the code with a higher point-obtention reliability that even if it still failed sometimes, helped mitigating the problem.

8.3.2 Time inconsistency

Another big problem when determining specific points is the inconsistency throughout time of the data provided by the frames. As this project is performed on a dynamic environment where images are continuously being captured and treated, not all the frames give the same information. This means that all the frames can suffer any type of alteration and thus they are not always providers of the whole truth. For this reason, is not viable or even realistic to process just one frame and treat the result given by that one frame as an absolute truth, as it can be considerably altered.

Consequently, and as precision is one of the key aspects for this project, one main countermeasure has been taken to prevent that measurement error from influencing the camera's performance. This method consists of taking advantage of the high amount of FPS obtained by analysing various frames to later use the median value (as seen on Image 55). This is particularly efficient in these cases as the high image acquisition speed allows treating a lot of frames in a small amount of time. Elseway, by taking the median values a precise measurement is ensured while avoiding the influence of some possible noisy data that could affect the mean value.

Image 55: Median of an array

This methodology has been applied to a high number of variables to ensure the maximum precision, as they could be the (X, Y) position of the object's centres, their angle or the position of the edges used to calculate the objects' size. But the most important one is the median used when measuring depth, as it previously made clear that this variable is the one with the noisiest measurements and even the higher number of empty values.

Thanks to this methodology, the impact of time inconsistency on the measurements has been significantly reduced. By leveraging the high frame rate and applying a median-based approach, the system can extract more reliable and stable values, minimizing the influence of noise and missing data. This ensures a higher level of precision in critical measurements such as depth estimation, object positioning, and size calculation. Ultimately, this strategy enhances the overall accuracy and robustness of the system, making it more suitable for dynamic environments where data fluctuations are inevitable.

9. Results

This section presents an overview of the outcomes achieved throughout the project's development, providing empirical evidence and observations of the implemented systems. It synthesizes the performance, capabilities, and identified limitations of the vision-guided robotic assembly cell, building upon the methodological details discussed in previous sections. The results demonstrate the project's success in meeting its core objectives while also highlighting areas for future refinement. Specifically, the subsequent paragraphs elaborate on the performance observed in critical areas such as object detection, 3D localization, hole identification, communication protocols, and real-time operational metrics.

The neural network model developed for object detection demonstrated robust performance. For instance, the instance segmentation model used showed a *98.98%* accuracy in mAP@0.5 metric. When considering the more demanding mAP@0.5:0.95, the model achieved *96.77%* accuracy, which is also a reliable result. The visual demonstration of the model is also a good representation of the achieved robust response, with appropriate bounding box/contour fitting, high confidence, and few detection errors like false positives, even handling semi-occluded objects more effectively. This model also showed an invariable performance for detecting objects under any lighting conditions, which is a clear indicator of its robustness and avoids the need for threshold adaptations of the model on different environments.

The system also demonstrated the capability of accurately localizing and characterizing objects in the 3D space, which is critical for robotic manipulation. Crucial information such as object size, rotation angle and optimal holding positions were calculated from the data extracted from the neural network's predictions and depth information. A filtering technique was also applied to manage the noisy depth data, enhancing measurement robustness. However, despite the good results obtained when extracting this information, the camera's accuracy limitation impeded a complete integration with the robot, as the camera measurement inaccuracy was determined to be about *±3.3mm*, which is outside the tool's *±2.5mm* tolerance. This means that if a full integration of both parts of the project is desired either the robot tool's tolerance must increase or further measures need to be taken to reduce the inaccuracy of the camera.

One of the best results of the project was obtained on the specialized module developed to determine the precise orientation of the 'finger' object. This module applies various filters and thresholds to identify the holes of the objects correctly while discarding all type of noise. These holes' contours are then converted to centimetres using a fixed ratio, reducing computationally intensive methods and while still ensuring accurate results. The orientation of the element is then determined based on the y-coordinate of the larger hole relative to the smaller one, a simplification valid due to the 'finger' object's consistent alignment with the camera's y-axis. This provides crucial information for the robot's mounting orientation which is completed with a *99,58%* accuracy.

Communication was established as a core component of the project, linking all hardware and processing units composing the system. High-speed and reliable data transfer from the camera to a local computer was achieved via a USB 3.0 connection. Subsequently, processed data was transmitted wirelessly using Wi-Fi to a separate computer controlling the robotic cell, leveraging threading structures for efficient and real-time

interaction. The communication operates on a command-driven basis, where the robot's computer sends specific commands to the local computer, triggering precise image acquisition and processing routines. Processed information is then compiled into a .JSON file and sent to the robot, enabling dynamic and informed robotic control. This robust communication framework ensures that image acquisition and processing occur only when explicitly requested, optimizing resource usage and system responsiveness. Apart from that, this communication method also enables the possibility of multi-device communication, which opens a framework for future capability expansions of the project such as the possibility of stablishing a centralized control of various workspaces simultaneously.

Finally, the system's real-time performance and throughput were key considerations in camera selection and system configuration. The Intel RealSense D455f camera, chosen for its overall depth accuracy and low noise, operates at up to 60 FPS. A balanced resolution of 640x480 pixels with a frame rate of 30 FPS was selected to ensure both sufficient image accuracy for processing and adequately real-time data flow. The use of a USB 3.1 connection was critical for maintaining consistent frame rates and avoiding frame loss experienced with USB 2.0.

Finally, the system's real-time performance and throughput were a key consideration in the system’s configuration. Here, a balance between resolution, minimum detectable distance, and FPS was seeked. A resolution of 640x480 was selected, offering a good balance by significantly reducing the minimum depth while maintaining high accuracy. The frame rate was chosen of 30FPS, providing an adequate balance between real-time data and processing precision. Using a USB 3.1 connection was also important in this aspect, avoiding frame loss encountered with USB 2.0. Furthermore, the command-driven image acquisition strategy used, optimizes resource usage by ensuring that frame capture and processing only occur when specifically requested by the robot, thereby contributing to efficient system throughput.

All in all, it can be concluded that the obtained results on the project show substantial achievements that show its reachability while also determining a clear path to follow for future progress if the project is wanted to be further on developed. Nevertheless, these aspects will be further analysed on the following “Conclusions” (Section 11) and “Future lines” (Section 12) chapters.

For a more complete overview of the achieved final results in the overall project, a demonstration video was also recorded, and it can be viewed on the Annex D, where both developing branches have been integrated together for a complete overview of the achieved milestones. This includes the current artificial vision path developed on this report and the robotic cell report developed by *Unai Amenabar*.

10. Economic report

This part of the report details the developing and implementation costs of the developed dual robot cell. Despite the global project being divided into two separate sections, if any company would like to purchase the product, both parts of the project would be included in the budget. However, the budget can actually be split into the engineering development costs and potential implementation costs.

Note that all costs and prices have either been provided by the university or companies that work with the Universidad Politécnica de Valencia.

10.1 Development costs

The development costs group all costs that have been made in the solution development phase, such as the design of the different tools, testing equipment and engineering personnel costs.

10.1.1 Personnel costs

Although the project was almost entirely carried out by the student, meetings were also held with other entities and university professors to clarify some doubts. Table 14 showcases the summarized total cost from people that has contributed to this project.

Table 14: Personnel costs

Person	Cost / Hour	Invested hours	Total cost
Student 1	30 €	480	14.400 €
Student 2	30 €	480	14.400 €
Promotor UPV 1	100 €	50	5.000 €
Promotor UPV 2	100 €	50	5.000 €
Promotor UGhent	100 €	10	1.000 €
Department technician 1	80 €	100	8.000 €
Department technician 2	80 €	25	2.000 €
Total		**1195**	**49.800 €**

10.1.2 Design costs

For the design of materials for later manufacturing, the university has provided the project team with a computer with SolidWorks license.

Table 15: Design costs

Element	Cost
PC	1.200 €
Monitor	200 €
SolidWorks license (students)	60 €
Total	**1.460,00 €**

10.1.3 Manufacturing costs

Considering that the costs of using the 3D printer are included in the cost of the department technician, the material cost can be calculated. Even the material costs can be broken down into different parts of the cell.

Table 16: Material costs of UR3e-2 tool

Element	Cost
Dowel pad	0,84 €
Support	1,03 €
Secondary pad	0,88 €
Bolts (M5x10)	1,73 €
Bolts (M4x10)	2,43 €
Total	**6,91 €**

As for the second tool design, as two options have been considered, two separated budgets can be made.

Table 17: Material costs of UR3e-1 tool

2nd tool (single chamber pad)

Element	Cost
Modular base	2,02 €
Pad holding wall	4,00 €
Link to coupler (tool side)	1,22 €
Inflatable pads (single chamber)	1,32 €
Suction cup limiter + guide	0,33 €
Tool - Coupling spacer	0,32 €
Bolts	6 €
Total	**17,21 €**

2nd tool (multi chamber pad)

Element	Cost
Modular base	2,02 €
Pad holder	0,33 €
Link to coupler (tool side)	1,22 €
Inflatable pads (multiple chamber)	1,14 €
Suction cup limiter + guide	0,33 €
Tool - Coupling spacer	0,32 €
Bolts	7 €
Total	**14,36 €**

Apart from the tools, the only element that the team has manufactured is the camera leveller.

Table 18: Material costs of camera leveller

Element	Cost
Leveller 1	0,57 €
Leveller 2	0,57 €
Bolts (M5x10)	2,88 €
Total	**4,02 €**

10.1.4 Material cost

As mentioned in the section *1.3 Background*, this project is based on the material lent by the systems and automation department, which includes all the wiring, robots' software and hardware and communication protocol systems. For the use of this materials, approximations have been made.

Table 19: Material costs of general use elements

Element	Cost
UR3e-1	15.000€
UR3e-2	15.000€
Intel RealSense *D455f*	500€
Electrical wiring	1.000€
Pneumatic circuit	750€
IKEA Oddvar	15€
OnRobot coupling	2.162,88 €
FESTO DHPS-20-A	694.77 €
Total	**35.198,88 €**

10.1.5 Total cost of development

Summing the previous costs, the total cost of development is calculated,

$$cost_{dev_{tot}} = 49800 + 1460 + 6.91 + 17.21 + 14.36 + 4.02 + 35198.88 = 86501.38[€] \quad l)$$

10.2 Implementation cost

If the logistical costs are not considered, the main implementation costs are the personnel, material and operator traineeship costs for cell assembly and fine-tuning process in the client's company.

10.2.1 Personnel costs

Table 20: Personnel cost at implementation

Person	Cost / Hour	Invested hours	Total cost
Assembly technician	50 €	40	2.000 €
Fine tuning engineer	100 €	100	8.000 €
Total		**140**	**10.000,00 €**

10.2.2 Material costs

The material costs are the same ones described in the development phase. In this case, for the second tool, the single pad design costs are only considered However, when selling the product to a company, a scaling factor is applied to obtain benefit from the sale. In order not to inflate the price too much, the scaling factor is set to 1,2. Therefore,

$$cost_{impl_{material}} = (6.91 + 17.21 + 4.02 + 35198.88) \cdot 1.2 = 42272.42[€] \quad m)$$

10.2.3 Total implementation cost

The total implementation cost sums the previous two costs and applies the VAT over it, which is 21% in Spain.

$$cost_{impl_{tot}} = (10000 + 42272.42) \cdot 1.21 = 63249.63[€] \quad n)$$

10.3 Project feasibility

Knowing the price tag of the developed product, the feasibility can be calculated by comparing the economic performance of the robotic assembler to the manual assembler. It can further be analysed how many robotic cells would be optimum for each operator.

10.3.1 Mechanical assembly technician

First, the mechanical assembly is done by each of the team members 5 times to determine the approximated assembly duration of an experienced technician. Image 56 shows the evolution of assembly duration for several tests.

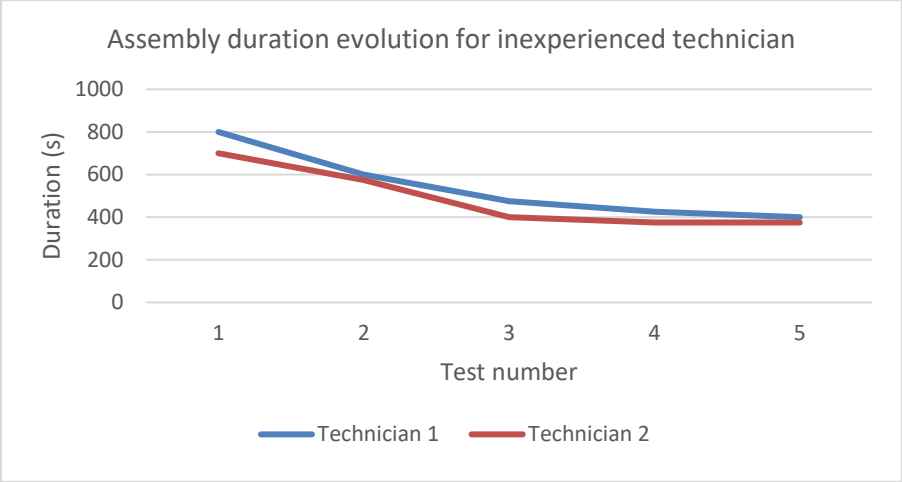


Image 56: Assembly duration evolution for inexperienced technician

Although in the first assemblies the duration may differ significantly due to competences of the technician or other different factor, the required time to do the assembly tends to converge. Considering this evolution, it is approximated that an experienced operator in furniture assembly could be able to perform the entire assembly in around 5 minutes. This time also includes the time required to select the objects of interest.

In Spain, the average wage of an assembly technician is approximately 10.15€ per hour *[42].* Adding the 15€ of material cost per chair, the total cost per chair can be calculated,

$$cost_{chair_{manual}} = \left(10.15\left[\frac{€}{\text{hour}}\right] \cdot \frac{1\ [\text{hour}]}{60\ [\text{min}]} \cdot 5\ [\text{min}]\right) + 15[€] = 15.845[€] \quad \text{o)}$$

However, it is also important to note that the manual assembly has higher productivity compared to the automated one.

$$\frac{chairs\ produced}{8\ hour\ shift/operator} = 8\ [\text{hour}] \cdot \frac{1\ [\text{chair}]}{5\ [\text{min}]} \cdot \frac{60\ [\text{min}]}{1\ [\text{hour}]} = 96\left[\frac{chairs}{shift/operator}\right] \quad \text{p)}$$

10.3.2 Automatic assembly

Considering that the average consumption rate of a UR3e is around 125 [W], plus the auxiliary systems, assumed to be around 100[W], sums up for,

$$P_{cell} = (125[\text{W}] \cdot 2) + 100[\text{W}] = 350[\text{W}] \quad \text{q)}$$

Which can be used to calculate the cost per chair of the dual robotic cell. The electricity cost in Spain in the moment this report has been done is at 0,14381[€/kWh], according to *Red-eléctrica* as for 26/06/2025 *[43].*

$$cost_{energy} = 350[\text{W}] \cdot \left(0.14381 \left[\frac{€}{\text{kWh}}\right] \cdot \frac{1\ [\text{kW}]}{1000\ [\text{W}]} \cdot \frac{1\ [\text{hour}]}{60\ [\text{min}]}\right) \cdot 27.5\ [\text{min}] = 0.023[€] \quad \text{r)}$$

And therefore, the total cost per chair,

$$cost_{chair_{automatic}} = 0.023[€] + 15[€] = 15.023[€] \quad \text{s)}$$

In the case of the automated assembly, productivity is significantly lower,

$$\frac{chairs\ produced}{8\ hour\ shift/cell} = 8\ [\text{hour}] \cdot \frac{1\ [\text{chair}]}{27.5\ [\text{min}]} \cdot \frac{60\ [\text{min}]}{1\ [\text{hour}]} = 17.45 \left[\frac{chairs}{shift/cell}\right] \quad \text{t)}$$

To reach the productivity of a manual assembly technician, multiple cells would be required to be monitored by an operator.

$$\frac{cells}{operator} = \frac{96}{17.45} = 5.5 \left[\frac{cells}{operator}\right] \quad \text{u)}$$

Considering that one operator can cope with 9 cells at a time, at maximum, being 6 the optimum, the implementation of a multi cell can be considered. However, it would require approximately eight times more space than the required by one operator. Even with cells sharing working space, the minimum working space would be six time more at minimum. This would increase the price of the cell indirectly, as the company would need to either relocate to a bigger industrial unit, enlarge the actual unit, or reduce the productivity.

10.3.3 Payback period

A valuable financial metric for evaluating the economic viability and efficiency of capital-intensive projects such as this one are the Return on Investment (ROI) and payback period. These metrics quantify the financial benefits of the project's implementation relative to the initial investment, indicating how long it will take for the generated savings to offset the upfront costs. Understanding the ROI allows stakeholders to make informed decisions regarding the strategic allocation of resources and the long-term profitability of such technological advancements.

To calculate the ROI and payback period for this automated assembly solution, we first need to determine the annual savings generated by the system. For this, an assumption that the facility operates for 2 shifts per day and 240 working days per year will be taken. So, by taking into account the previously calculated fabrication costs for a chair using both methods, a direct calculation of the chair production's cost-saving can be calculated.

$$\frac{saving}{unit} = cost_{chair_{man}} - cost_{chair_{auto}} = 15.845[€/u] - 15.023[€/u] = 0.82[€/u] \quad v)$$

If we assume that the automated system operates with 7 cells, and knowing that it exhibits a productivity of 17.45 chairs per shift per cell, its annual production can also be calculated.

$$Anual\ production = 7 \cdot \left(17.45\left[\frac{u}{shift \cdot cell}\right] \cdot 2\left[\frac{shift}{day}\right] \cdot 240\left[\frac{days}{year}\right]\right) = 58632\left[\frac{u}{year}\right] \quad w)$$

Based on this production volume, the total annual savings from implementing the automated cells can be calculated:

$$Anual\ savings = Anual\ production \cdot \frac{saving}{unit} = 58632\left[\frac{u}{y}\right] \cdot 0.82\left[\frac{€}{u}\right] = 48299.1\left[\frac{€}{y}\right] \quad x)$$

This represents the recurring financial benefit obtained each year from the operational efficiency of the automated system compared to manual methods. But in order to know if this savings are enough, it must be compared to the total investment cost.

These costs can be divided into 2 categories:

- Development costs (inherent to the number of cells implemented): 86,501.38 €
- Implementation costs: 63,249.63 €/$cell$

So, for this scenario where just 7 cells will be implemented, the total cost of the investment would be the following:

$$Cost_{inv} = cost_{dev} + cost_{impl}$$
$$Cost_{inv} = 86{,}501.38[€] + (7[cell] \cdot 63{,}249.63\left[\frac{€}{cell}\right]) = 529{,}248.79[€] \quad y)$$

If more cells would be implemented the development costs would be less noticeable, but as the final result would not suffer significant differences, it has been taken into account. Therefore, the project will have the following Payback period to recover the investment:

$$Payback\ period = \frac{Cost_{inv}}{Anual\ savings} = \frac{529{,}248.79[€]}{48299.1[€/year]} = 10.96[year] \quad z)$$

10.3.4 Conclusion

As it can be seen on the ROI calculation, almost 11 years would be needed to recover the investment and for starting to be profitable. These are remarkably big numbers that are not attractive at all for the clients. A project is considered attractive when its payback period is about 3-5 years, which leave this project far away from the goal *[44]*. So, although the developed product supposes an improvement in productivity and cost over previous solutions, it is still not feasible for a company. To match or improve the efficiency of a manual assembly technician, the automated assembly must decrease the required time significantly, increasing drastically its productivity. For so, the use of auxiliary systems or additional robots should be re-explored.

11. Conclusions

To conclude with the project, a profound analysis of the objective's completion must be done. The main objective of this project was to successfully achieve a 3D assembly of randomly placed elements within a predefined working area, utilizing two collaborative *UR3e* robots guided by a 3D *Intel RealSense* camera for real-time analysis and inspection. Taking that into account, it can be said that the project has demonstrably achieved his main goal and has proven its viability despite facing various challenges and acknowledging areas for future development. So, even if not all the intended tasks were correctly developed, the obtained results have established a robust enough proof-of-concept for automating complex assembly processes, yielding valuable results and laying a strong foundation for a possible adaptation to real-life industrial applications, thereby holding significant potential to reduce production cost and complexity.

By looking at the results obtained on the Section 10 (Economic report), it can be said that the main goal of reducing the production time, cost and complexity a 50% was not completely met, even if several advances were obtained. The complexity reduction may be the part where more advances were made. Here, it was achieved to complete the same task that was previously done by 7 operators with a single one, having this one to perform a slightly more complex task but physically less demanding. The cost reduction was also significant, reducing the annual cost for the enterprise on about 48299,1€. This is quite a remarkable result, but due to the massive initial investment needed to implement the solution, the project remains unattractive to customers, resulting on 10,96 years' return of investment. This was due to the low productivity of the automated cells. These cells didn't manage to accomplish the stablished goal of reducing the production time on a 50%, but they rather increased it on a 550% going from 5 minutes to 27,5 minutes for mounting a chair. Therefore, it can be clearly seen that the time reduction was one of the main problems on the project and if future developments are going to be performed, the main focus should be on this time reduction by applying new or better mounting methods.

For a deeper acknowledgment of the objective's completion, each of the main defined tasks could be individually analysed to see their individual performance level achieved.

For example, the crucial initial step of selecting the camera placement method and optimizing its position within the working environment was successfully executed. The camera was strategically positioned above the assembly area, providing the essential visual input for the system. This foundational setup proved to be effective in offering a comprehensive view of the workspace, enabling all subsequent visual analysis and robotic guidance functions.

Another task which demanded considerable efforts was the camera system's environmental tuning and noise suppression. The camera was adjusted to the specific surrounding conditions, with the aim of suppressing external noise and accurately identifying assembly components. So, while the chosen camera and environmental setup allowed for successful identification of elements, the inherent characteristics of the camera influenced the ultimate level of precision. Even if the precision was good, it was not enough for the direct integration with the robot's picking algorithm. Nevertheless, this system demonstrated a robust capability to discern objects, contributing directly to the successful execution of the main objective.

A critical enabler task for the project's primary goal was the successful implementation of 2D object identification. By using a NN capable of recognizing predefined assembly elements, the system effectively located and categorized the components with a high accuracy. This fundamental capability was vital for the robots to accurately perceive and interact the individual parts and initiate the assembly process.

While the initial aspiration included a deeper integration into a comprehensive 3D model for advanced object understanding (such as full 3-axis rotation and complete 3D Neural Network segmentation), the project strategically prioritized achieving the core 3D assembly objective. Although a full 3D NN was not implemented due to computational incompatibilities, the system effectively leveraged the 3D depth data of the camera to extract essential information such as object position and basic orientation, which was crucial for the 3D assembly. The generation of a full 3D point cloud also provided a valuable spatial context, demonstrating the system's ability to operate within a 3D workspace. So, even if it at the moment is just used for visualization due to significant performance impact (reducing streaming rates from ~6 FPS to ~0.2 FPS), this partial 3D integration is still valuable on the way towards a 3D assembly objective.

A successful outcome that directly contributed to the main objective was the system's ability to calculate the optimal holding point and orientation for each assembly element. This precise positional and orientational data was accurately transmitted to the UR3e robots, enabling them to execute precise pick-and-place operations. This seamless integration between vision and robotic control is a fundamental capability, indispensable for a successful 3D assembly.

A significant achievement that directly addressed the "randomly placed elements" aspect of the main objective was the development of an algorithm for defining the picking order. This algorithm not only facilitated assembly following the predefined sequence, but was also improved to operate effectively with randomly placed initial elements. This allowed the system to intelligently identify the most trustworthy objects that were reachable for the robot and based on that information decide what the robot’s next move should be. This includes detecting which object is subsequently needed on the assembly process by the robot and depending on the available elements detected by the camera directly use the one with the highest confidence out of the detected ones or store them in the warehouse to uncover the object placed beneath them. This achieved capability is essential for automating multi-component assembly sequences, contributing directly to the optimal procedure specified in the main objective.

While the ultimate objective of examining assembly results for quality control and defect identification was a crucial goal for ensuring an optimal procedure, this particular task could not be developed within the project's timeframe. Nevertheless, its absence does not diminish the successful completion of the core 3D assembly objective. Furthermore, this area represents a clear path for future development, which would further refine the system’s capability for real-world industrial applications by providing crucial quality assurance procedures.

12. Future lines

Looking ahead, this project presents several exciting possibilities for a hypothetical continued development. While the current system provides a robust foundation for automated object manipulation, future efforts will focus on expanding its capabilities, improving its performance and integrating more advanced technologies to explore new applications for robotic manipulation and computer vision. On that account, the following lines aim to address current limitations and unlock new possibilities for the project's application.

One of the main problems that limited most the capabilities of the project was the camera's performance itself. This camera was really cheap compared to other competitors and offered enough quality for this specific project. Nevertheless, it also has quite a lot of limitations that generated some challenges along the way, reducing the model's performance and also limiting some interesting processing paths that could be very helpful to extend the project's capabilities. So, replacing this camera by a better one would suppose a crucial update that would help improving the model's accuracy and reliability. This improvement would not only improve the performance of both hole and object detection functions but also would create a more accurate point-cloud with a higher resolution that could be used to implement 3D segmentation methods. In conclusion, even if changing to a superior camera would mean a considerable increase in costs, it would suppose an even higher increase in performance and scalability of the project.

Developing robust error detection mechanisms is another crucial future line. Due to time constraints, this aspect could not be developed, but it would be crucial for the project's scalability and its implementation on real production lines. This error detection could be implemented as a quality check after assembly. To facilitate this task, additional cameras could be strategically placed in near the assembly zone, either placing it directly above the station or in any other position that provide alternative perspectives of the assembly space. These extra cameras would make it easier to identify assembly errors without needing to move the object to the object-placing zone, improving efficiency and reducing the production time. Furthermore, this multi-camera setup offers an opportunity to monitor the assembly process itself as well, identifying inefficiencies or errors during the process that could be stored and later used by the operator for improving to the robot's program and overall operational efficiency.

Moreover, as previously mentioned, transitioning from the current 2D Neural Network segmentation method to a 3D one would mean a significant improvement. This would require meaningful changes in the computer, as at the moment is not able to handle the 3D-NN libraries. However, the benefits of this change would be noticeable, as a 3D approach provides a much deeper understanding of the environment and leading to more precise segmentation methods. For example, this approach would enable the detection of all objects in the workspace no matter if they were above or beneath the others, unlike the current approach that only allows upper object detection. Furthermore, a 3D segmentation method would also allow to detect objects that were rotated across all three axes, rather than being limited to the current planar Z-axis rotation. This evolution would unlock far more complex detection and manipulation possibilities that would notably increase the projects reach. Finally, if this approach was coupled with a better camera, the quality

of the 3D environment would be greatly enhanced, allowing for the extraction of even more detailed information directly from the point-cloud that would come out handy for the point-cloud's processing.

For a significant boost in productivity, a key future development could also be the integration of multiple simultaneous workspaces. This expansion of capabilities would involve additional robotic cells with their additional workspace camera each, increasing the costs significantly. This could not only help being more profitable due to the proportional increase in production but can also help reducing the costs as a lot of elements would be shared between workspaces. Crucially, all the cameras would be controlled by a single, centralized local server running the same core code. Each robot would then connect to this server, requesting information specifically from its designated camera. This centralized control would streamline operations, manage resources efficiently across multiple stations, and enable parallel assembly processes. Thanks to that, a single operator would be able to control all the workspaces at the same time, as a single computer will be able to run them all. This is also interesting because the operation of mounting a single chair takes around 15 minutes while the operator is only needed for 1-2 minutes in the entire process. This leaves the possibility of the operator controlling up to 10 workspaces simultaneously and using a single computer, which would mean a huge increase in productivity and cost efficiency of the capabilities.

Finally, adding an automatic assembly detection feature based on SolidWorks or AutoCAD assemblies would also be a meaningful step to follow on the project's development, considerably increasing its adaptability and scalability. This would involve integrating CAD models directly into the vision pipeline. By doing so, the object recognition module could automatically adapt to new elements of a new assembly design, eliminating the need for extensive manual retraining or configuration whenever a new product or component is introduced to the production line. This would drastically reduce setup times and allow for rapid deployment in diverse manufacturing scenarios, making it one of the best possible steps forward to take on the project.

Personal assessment

This report of the "DESIGN, DEVELOPMENT, AND EVALUATION OF A FLEXIBLE MANUFACTURING CELL USING VISUAL GUIDANCE FOR COLLABORATIVE ROBOTS FOR AUTOMATIC ASSEMBLY AND MOUNTING OPERATIONS" master dissertation develops the following concepts in the text, correctly justified and discussed, centred on the field of Computer vision for industrial applications:

Table 21: Personal assessment

Concept (ABET)	Achieved? (Y/N)	Where? (Pages)
1. IDENTIFY:	Y	1-10
1.1. Problem statement and opportunity	Y	1-4
1.2. Constraints (standards, codes, needs, requirements & specifications)	Y	4-5, 9-10
1.3. Setting of goals	Y	5-6
2. FORMULATE:	Y	13-21
2.1. Creative solution generation (analysis)	Y	13-14, 17-20
2.2. Evaluation of multiple solutions and decision-making (synthesis)	Y	15-16, 20-21
3. SOLVE:	Y	77-78, 80-81
3.1. Fulfilment of goals	Y	80-81
3.2. Overall impact and significance (contributions and practical recommendations)	Y	77-78

Bibliography

[1] European Commission (October 1st, 2024). *What is Industry 5.0?*. Link: https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en

[2] Jean-Philippe Raiche (December 14th, 2022). Proaction Internation. *A new phase of the industrial revolution.* Link: https://blog.proactioninternational.com/en/industry-5.0-the-next-industrial-revolution-is-people-centric

[3] Rocío González, CincoDías (December 30th, 2022). *El 34% de las pymes no tiene previsto invertir en tecnología*. Link: https://cincodias.elpais.com/cincodias/2022/12/22/pyme/1671743897_122574.html

[4] Ministerio de Industria y Turismo(n.d.). *Statistics on smes and publications.* Link: https://industria.gob.es/en-us/estadisticas/paginas/estadisticas-y-publicaciones-sobre-pyme.aspx

[5] Lou DiLorenzo, et al. Deloitte (October 13th, 2023). *From tech investment to impact: Strategies for allocating capital and articulating value*. Link: https://www2.deloitte.com/us/en/insights/topics/leadership/maximizing-value-of-tech-investments.html

[6] Huayan Robotics (March 8th, 2023). *Han's Robot Application - Furniture Assembly Solution* [Video]. Link: https://www.youtube.com/watch?v=v4-3NK_g_Rc&ab_channel=HuayanRobotics

[7] NBC News (April 30th, 2018). *This Robot Can Assemble Your Ikea Furniture | Mach | NBC News.* [Video] Link: https://www.youtube.com/watch?v=R2I5pstEtD8&ab_channel=NBCNews

[8] Automated Assembly Systems, ASSEMBLY (May 2nd, 2018). *Robotic System Autonomously Assembles an IKEA Chair*. Link: https://www.assemblymag.com/articles/94283-robotic-system-autonomously-assembles-an-ikea-chair

[9] COEX (August 26th, 2021). *Robotic furniture assembly* [Video]. Link: https://www.youtube.com/watch?v=hGOejYBf41U&ab_channel=Ji-HunBae

[10] IKEA (n.d.). *Oddvar*. Link: https://www.ikea.com/es/es/p/oddvar-taburete-pino-20249330/

[10] United Nations: Department of Economic and Social Affairs (n.d.). *The 17 goals*. Link: https://sdgs.un.org/goals

[11] Hilti (n.d.). *Exoesqueleto de hombro EXO-S.* Link: https://shorturl.at/kUtm4

[13] Quo (January 12th, 2015) *Exoesqueleto Hal*. Link: https://quo.eldiario.es/tecnologia/g43742/exoesqueleto-hal/

[14] Straits Research (April 23rd, 2023). *Military Exoskeleton Market*. Link: https://shorturl.at/2h1aH

[15] Apple (n.d.). *Apple Vision Pro*. Link: https://shorturl.at/5s1bk

[16] chirag, Appinventiv (September 26th, 2024). *The Costs and Benefits of Developing an AI-Powered Smart Personal Assistant App.* Link: https://shorturl.at/vJ6bP

[17] Raghava Kashyapa, Qualitas Technologies (February 23rd, 2023). *How Expensive are machine vision solutions*? Link: https://shorturl.at/0KenN

[18] ITRex, Hackernoon (October 23rd, 2023). *Machine Learning Costs: Price Factors and Real-World Estimates*. Link: https://hackernoon.com/machine-learning-costs-price-factors-and-real-world-estimates

[19] Devicebase (n.d.). *ABB IRB140.* Link: https://devicebase.net/en/abb-irb-140

[20] ABB (n.d.). *IRB1100.* Link: https://new.abb.com/products/robotics/robots/articulated-robots/irb-1100

[21] Universal Robots (n.d.). UR3e. Link: https://www.universal-robots.com/es/productos/

[22] Hansard, M., Lee, S., Choi, O., & Horaud, R. P. (2012). *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media.

[23] Patel, D. K., Bachani, P. A., & Shah, N. R. (2013). *Distance measurement system using binocular stereo vision approach.* Int J Eng Res Technol, *2*(12), 2461-2464.

[24] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. *Depth mapping using projected patterns*. Technical report, U.S. Patent No. 20100118123, 2010.

[25] Scantech. (February 1st, 2024). *What is structured light 3D scanning?* Link: https://www.3d-scantech.com/what-is-structured-light-3d-scanning/#:~:text=Structured%20light%20scanning%20employs%20a,points%20onto%20an%20object%20sequentially

[26] Intel® RealSense™ (July 2023). *Intel® RealSense™ Product Family D400 Series datasheet*. Datasheet Number: 337029-016

[27] Intel RealSense. (September 24th, 2024). *Compare Intel RealSense Depth Cameras (Tech specs and Review) – Intel® RealSense™ Depth and Tracking Cameras*. Intel® RealSense™ Depth and Tracking Cameras. Link: https://www.intelrealsense.com/compare-depth-cameras/

[28] Rutronik Elektronische Bauelemente GmbH. (n.d.). Intel® RealSenseTM Stereo Depth V 3.0. In *Intel® RealSense™ Product Overview*. Link: https://www.rutronik.com/fileadmin/user_upload/IntelRealSense_Digi__EN_.pdf

[29] Intel. (July 2023). *Intel® RealSenseTM Product Family D400 Series Datasheet*.

[30] A. J. Sánchez. Universidad Politécnica de Valencia (2025). *Lightning systems in image acquisition*. Course: Artificial Vision

[31] Grunnet-Jepsen A. et al. Intel RealSense. (March 2022). *Intel® RealSense™ Self-Calibration for D400 Series Depth Cameras*. Link: https://dev.intelrealsense.com/docs/self-calibration-for-depth-cameras

[32] Y. Fernández. Xataka (February 28th, 2020). *USB 3.0: qué es y cuáles son sus diferencias respecto a USB 2.0*. Link: https://www.xataka.com/basics/usb-3-0-que-cuales-sus-diferencias-respecto-a-usb-2-0

[33] CCI Robotics. (September 16th, 2019). *Transformation Matrices for Object Motion* [Video]. YouTube. Link: https://www.youtube.com/watch?v=dpqNZmXLHYI

[34] R. Fisher et al. W3C (2023). *Closing*. Link: https://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm

[35] S. Sarker et al. Springer-Verlag GmbH Germany (May 18th, 2024). *A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation*. Link: https://www.researchgate.net/publication/380696927_A_comprehensive_overview_of_deep_learning_techniques_for_3D_point_cloud_classification_and_semantic_segmentation

[36] Lyzr (April 14th, 2025). *Batch Size*. Link: https://www.lyzr.ai/glossaries/batch-size/#:~:text=Smaller%20batches%20can%20provide%20more,but%20may%20require%20more%20memory.

[37] A. Firdauz et al. Universiti Teknologi Malaysi (October 21st, 2019). *Review on Techniques for Plant Leaf Classification and Recognition*. Link: https://www.researchgate.net/publication/336707258_Review_on_Techniques_for_Plant_Leaf_Classification_and_Recognition

[38] A. Pushkar. IntelliPaat (November 23rd, 2024). *Backpropagation Algorithm in Neural Network*. Link: https://intellipaat.com/blog/tutorial/artificial-intelligence-tutorial/back-propagation-algorithm/

[39] Deng X. Et al. Information Sciences (February 15th, 2016). *An improved method to construct basic probability assignment based on the confusion matrix for classification problem*. *Link:* https://doi.org/10.1016/j.ins.2016.01.033

[40] Olalekan J. et al. Nigerian Defence Academy (2023). *Genomic data science systems of Prediction and prevention of pneumonia from chest X-ray images using a two-channel dual-stream convolutional neural network*. Link: https://www.sciencedirect.com/science/article/pii/B978032398352500013 6?via%3Dihub

[41] Doxygen. OpenCV (June 4th, 2025). *Contour features*. Link: https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html

[42] Indeed (2025). *Sueldos por hora de Montador/a de muebles en Faster Empleo ETT en España.* Link: https://es.indeed.com/cmp/Faster-Empleo-Ett/salaries/Montador-a-de-muebles

[43] esios red eléctrica (June 26th, 2025). *Término de facturación de energía activa del PVPC*. Link: https://www.esios.ree.es/es/pvpc

[44] L. Bakić. The Productive Company Inc. (February 20th, 2025). *Payback Period in Project Management: Formula & Examples*. Link: https://productive.io/blog/payback-period-in-project-management/

Annex C

[1] Cambridge Conservation Initiative (n.d.). *"Sustainable Development Goals Tool," SDG Tool*. Link: https://sdgtool.com/

Annex A: Project Planning

One of the most important tasks on a project is to make a proper planning as it will be the guide and base of management during the whole project. On this part, the process to get to the initial planning is explained as well as the modifications and decisions taken from the point of view of management of the project to end up with the real tasks division followed to achieve the objective.

1. Initial planning

The initial planning is the planning made at the beginning of the project, that will be used as the base of the whole project as it is used to be a guide and a reference to control the project. To make a proper planning, some steps must be followed according to the PMI methodology shown on Image 57. PMI is the acronyms of Project Management Institute, the biggest and the principal worldwide organization on terms of project management. 1. On this case, the project does not start from zero, there are already from the beginning stakeholders and the information collected, given a problem solution proposal.

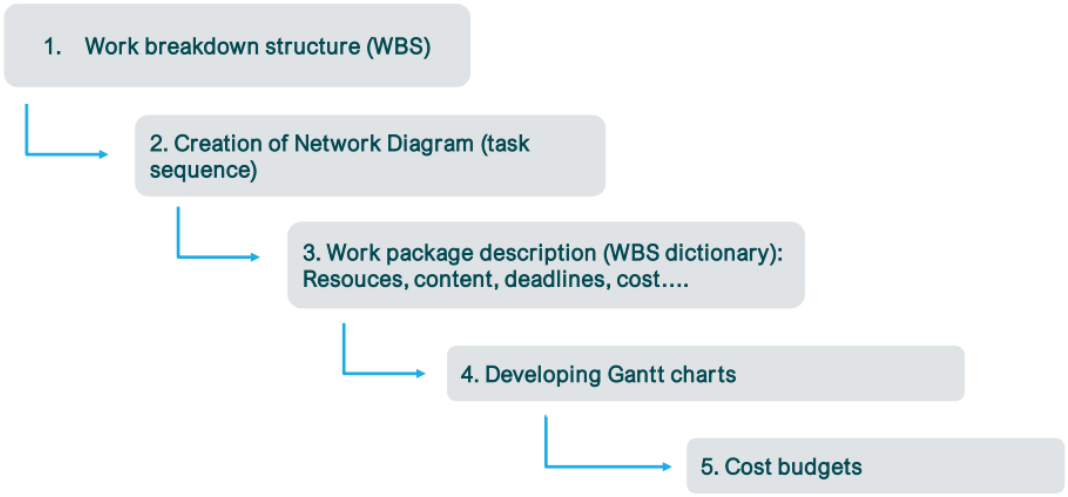


Image 57: PMI methodology steps

12.1 Objectives

From the given proposal and knowing the project requirements, to work all in the correct direction, the main objective as well as the minor grade objectives were defined. This objective clearly defines the cost, timelapse and specifications of the project while smaller milestones specify what needs to be done on track towards the main target. The S.M.A.R.T objectives of this project can be found on the Section 1.4. Moreover, they have been agreed with project tutor in order to avoid misunderstandings and deviations.

Once the objectives have been properly defined, a project plan has been created following the PMI methodology.

[1] https://www.pmi.org/

12.2 Work breakdown structure

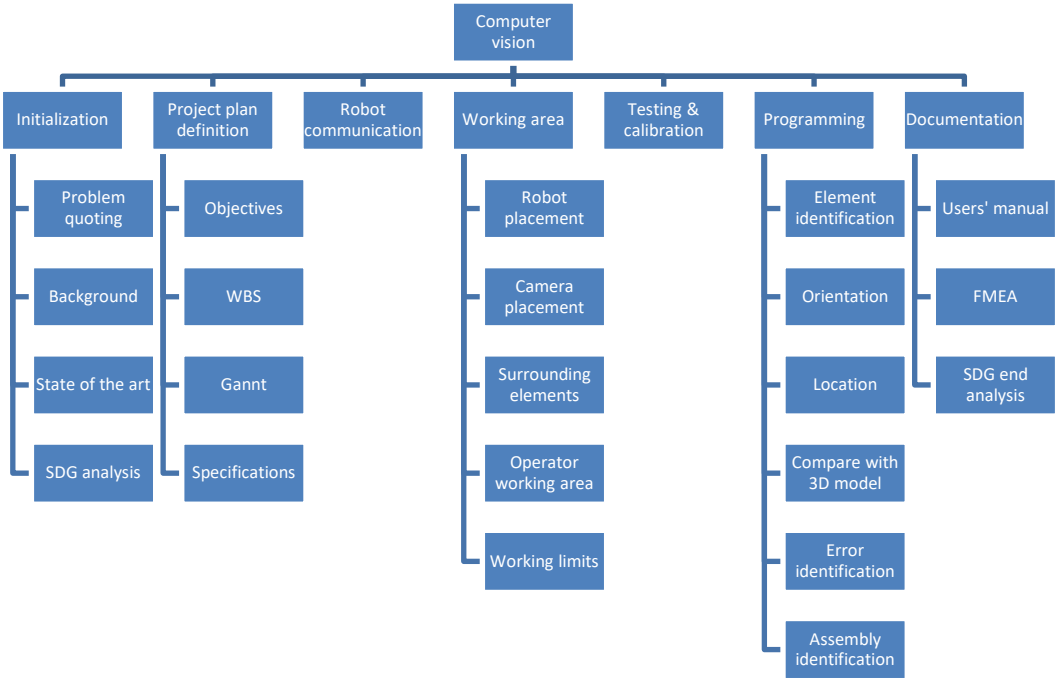


Image 58: WBS of the project

As first step of this methodology, a WBS has been created, which is a hierarchical decomposition, deliverable-oriented, of the work to be done on project. 2.That means that the Work Breakdown Structure is a tree diagram in which the project is broken down into the smallest divisible parts that help to make the work more manageable and approachable. These parts are classified within deliverables and sub deliverables, which help clarifying the projects' tasks in order to achieve the project objectives in the faster and best way possible. With this WBS, the project is depicted in a visual way on the Image 58, limiting the project scope boundaries.

12.3 Network diagram

On the WBS, the working packages that are necessary to complete the project have been identified. Once these activities have been stated, they were ordered considering which activity needs to be carried out before or after the others or if they can be done without any predecessors. All those activities have been linked in a network diagram or pert chart, showing each one's predecessor and successor activities. 3. Elseway, an estimation of each activity's duration has been made to complete it, determining which activities where critical and which of them had some slack.

On the following page, the whole network diagram can be seen on the Image 59.

[2] https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883#
[3] https://www.pmi.org/learning/library/pert-cpm-network-precedence-diagram-3736

Network Diagram

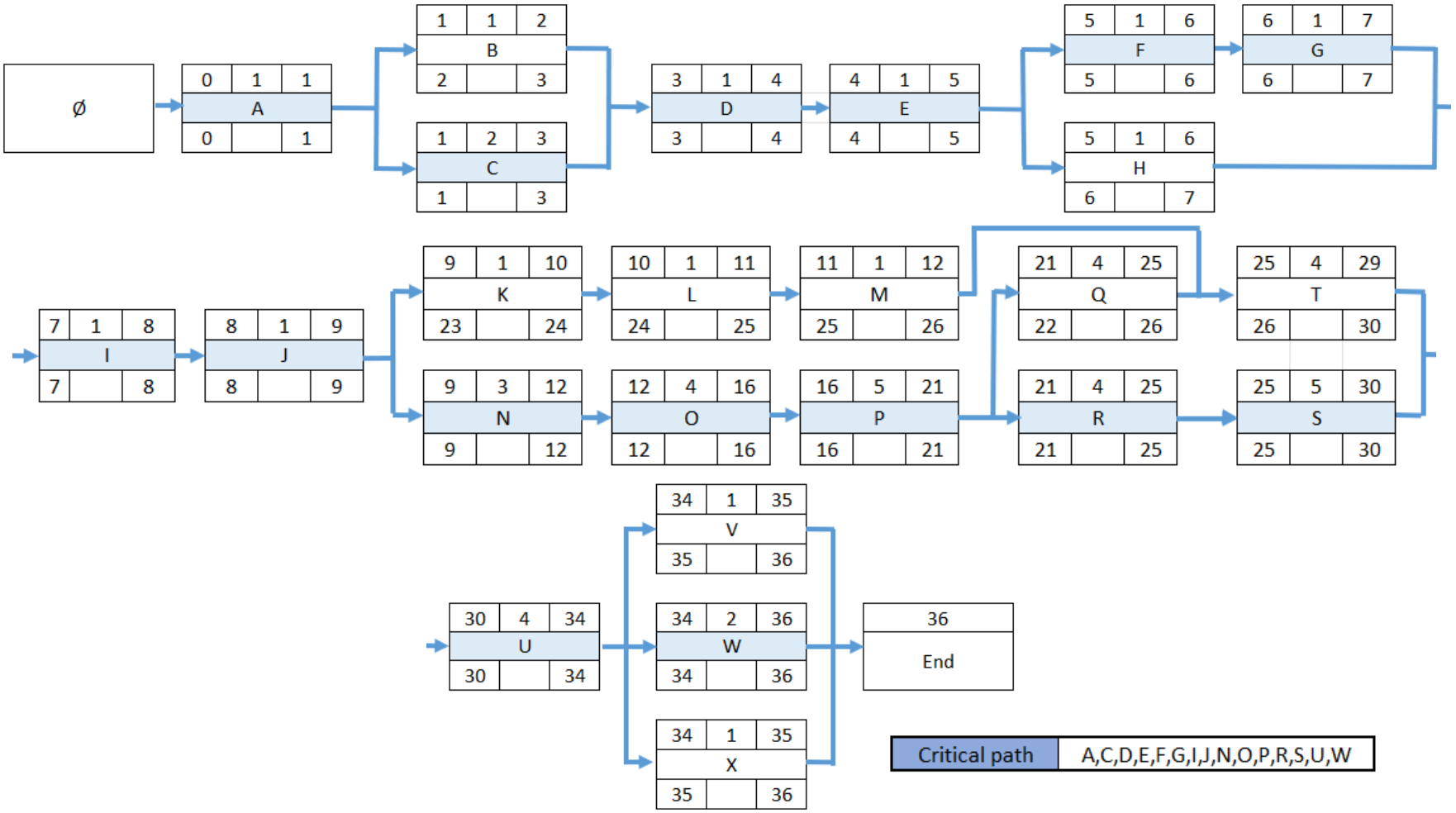


Image 59: Network diagram

On the network diagram above, apart from visualising how the tasks are connected between them, the critical path can be also defined. The critical path is composed of activities that cannot be delayed with respect to the initial plan, or so the whole project will be delayed. Critical activities are those that have no slack, that´s it, the early and late start and finish match. On this project, the critical path is composed by the following activities: A-C-D-E-F-G-I-J-N-O-P-R-U-W. Each of the activities being:

- A → Problem quoting
- C → State of the art
- D → SDG analysis
- E → Objective definition
- F → WBS + Network diagram
- G → Gantt chart
- I → Robot placement
- J → Camera placement
- N → Camera calibration
- O → Element identification
- P → Orientation
- R → Compare with 3D model
- U → Communication with robots
- W → FMEA

These activities cannot be delayed so they will be provided with more resources and higher priority. If any of this activity is delayed, all the resources are going to be used to advance so as there is no slack on the deadline.

12.4 Resources

The resources of this project can be classified on two, human and material resources. Both play an essential role on such a concentrated project where there is no time to go back or make changes during the project.

When it comes to the human resources, as there is just 2 people doing the project, effort made by the developer of the project is measured. This means that as the workload will sometime be higher, the person in charge would need to do an extra effort to complete the tasks in time or even ask to his companion for help. But both people being concentrated on just one task of this project could mean leaving a critical task of the other project to be left undone, needing a balance between both, so that none of them is delayed.

There are also two more people that can help the development of the project go straight. The first one is the *UPV* project tutor, who oversees the quality and timing of the tasks and the project's overall development focusing on the technical aspects. The other person is the *UGent* tutor, who follows the project continuously keeping track of the project's evolution, as well as providing also with extra expertise knowledge.

When it comes to material resources there are some specific materials in which the project will depend to complete certain tasks. Most of the tasks can´t be done without the correct material. Luckily, this material most of the times is exclusively used for this specific project, but sometimes other people will also need to use the same material. This means that even if we are on a critical stage where we have all the necessary to begin with a task, this task

cannot be developed without this material. These critical materials are mainly the assembly elements, the 3D camera and the collaborative robots themselves. Some other materials can also be demanded by different tasks, so if one specific material such as the computer is being used for one task, other tasks that need this element can´t be developed until the first one is paused or completed, even if they are parallel events on the network diagram. Fortunately, there is no time limitation with those materials as most of them are independent for the machine, but as some elements must be shared with other people an agreement of usage must be achieved between the interested parts.

12.5 Initial Gantt

Finally, considering everything that has been developed previously, a Gantt diagram has been designed where the tasks are divided into weeks, taking into account the milestones, the duration and sequency of the tasks stated on the network diagram and the resources needed by each of the tasks.

In a Gannt diagram, project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, as well as dependencies, scheduling and deadline. This is useful to keep tasks on track when there are multiple stakeholders. Project management solutions that integrate Gantt charts give managers visibility into the project's workloads, as well as current and future availability, which allows for more accurate scheduling. 4.

In this chart, the point where the project should be, or at which stage each task should be according to the initial planning can be seen in a very visual way. The rows are used to show the activities to be conducted during the project and in the columns the weeks of the year in which each task will be held. Predictions are made of when each activity is expected to start and finish. Thanks to this, the whole project is represented in a very visual way.

Usually, this chart needs to be compared with the available resources at each moment. As the resources requirements of each task can enter into conflict with another one that demands the same resource on the same time instant. For this reason, a workload distribution chart is typically drawn for the initially proposed Gannt chart, to then level the resources and with that adapt the Gannt chart. This valances out the workload of the project and also makes the planning easier to follow.

Even so, for this specific case is not necessary to do any of these modifications, as the initial workload chart is already valanced and the resources correctly distributed. Therefore, the Gannt will also be kept as on the initial planning, giving as a result the initial Gantt that can be found on the Section 1.6.

[4] https://www.apm.org.uk/resources/find-a-resource/gantt-chart

Annex B: Supplementary Code Documentation

This annex provides detailed documentation for code modules and functionalities that are no longer actively used in the main program but were part of the project's development. It also includes explanations for auxiliary scripts and functions that play secondary roles and were not extensively detailed in the main report. Understanding these components offers insight into the project's evolution and the rationale behind certain architectural decisions.

2. Streaming loop

The original main program operated as a continuous streaming loop. In this previous iteration, the system continuously acquired and processed images from the camera at a set frame rate, regardless of whether the robotic cell required new data. This meant that image analysis was performed constantly, consuming processing resources even when idle. This approach was later replaced by the current command-driven system, where image capture and processing only occur upon specific instructions from the robotic cell, optimizing resource usage and data flow. The code for this continuous streaming loop is no longer active within the main program.

The program began by importing necessary libraries for camera interaction and image processing, along with initializing global variables. The *Intel RealSense* camera was also configured for colour and depth streaming. Then, *OpenCV* windows were set up for interaction and frame alignment objects were initialized, which are crucial for pixel correspondence between frames. The last step performed before starting the streaming loop was the discarding of the first 20 frames to avoid noise and ensure stable sensor readings.

The core of the program consisted of a while loop, continuously attempting to acquire frames from the camera with a timeout mechanism. Once the frames were obtained, they were aligned to the colour image, ensuring the accurate correspondence between pixels of both frames. Afterwards, both aligned and unaligned frames were saved for possible later uses and the intrinsic parameters of the camera were extracted. But before that a validation step was implemented to check for missing or corrupted frames, skipping the current cycle if issues were found. The raw frame data was then converted into NumPy arrays, for their posterior usage.

Another extra feature that this code had and was later discarded on the newer versions of it, is an interactive depth measurement system. This feature consisted of measuring the depth of the pixel where the clicked on any of the frames. This point was then displayed on both the frames alongside its depth for the user to see, as it can be seen on the Image 60.

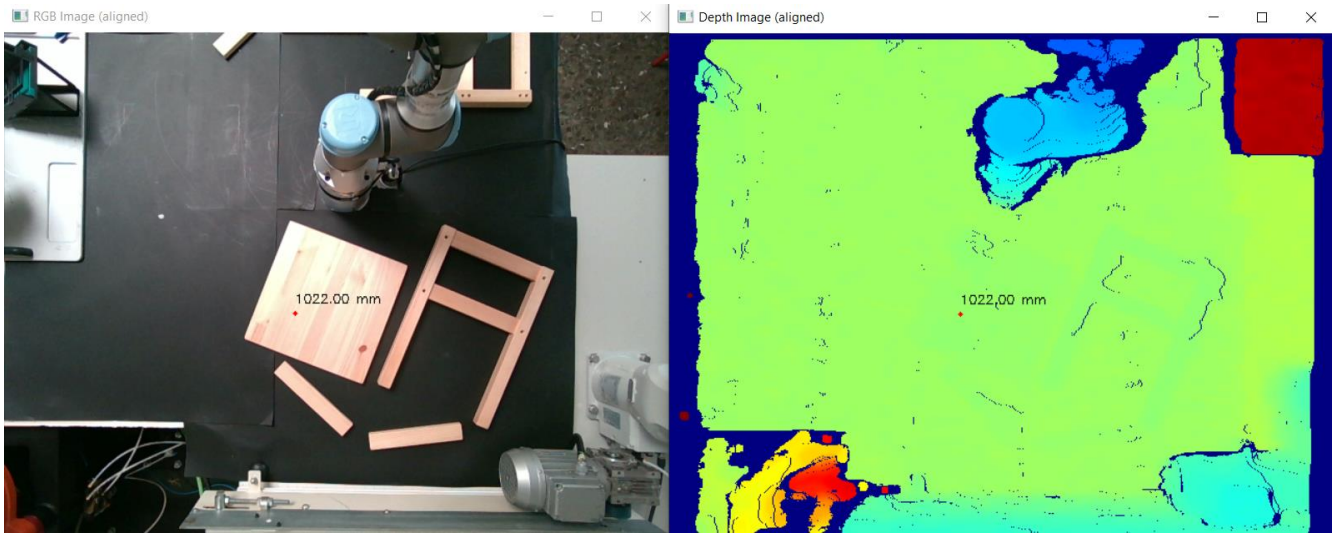


Image 60: Mouse-click based pixel analysis

Beyond these core functionalities, the streaming loop also contained various image processing sections, representing the different image processing options with their respective function each. These functions can be divided by their specific task:

- **Hole Identification:** The "Identify_holes" function detects and determine the orientation of the 2 holes of a finger object. It is the same function that is still in use, explained on the Section 7.2.
- **Object Identification**: The "identify_objects" function was used for coplanar object detection, using a manual segmentation approach. This is the one explained afterwards in the annex.
- **Object Identification (Depth-based)**: On the other hand, the "identify_upper_objects" function provided a more specialized object detection method, using depth information to focus just on the upper elements, but still using the same "identify_objects" function.
- **Camera Calibration (Transformation Matrix):** From this code also, the camera calibration code could be activated. This involved triggering the process explained on the Section 5.4.
- **Point Cloud Generation:** The "generate_point_cloud" function creates 3D point clouds from the captured colour and depth data. Originally intended for developing 3D Neural Networks, it is now used for pure visualization purposes. This function will be explained afterwards in the annex.
- **Image Buffering/Saving:** This part of the code does not serve as an image processing step, but more as a image storing part. This was used to create different datasets with either processed or unprocessed images, facilitating data collection and debugging.

Finally, the loop would break anytime the ESC key was pressed. Upon exiting the loop, the image pipelining would stop, and all displayed windows would close, releasing the system's resources.

3. Light adaptation code

The light adaptation code was initially developed to manually adjust the segmentation thresholds for both object detection and hole segmentation. This was achieved through a user-friendly interface that allowed manual selection of thresholds before the main program's execution. This ensured that the system could

adapt to varying lighting conditions, a crucial factor for accurate image processing. While the object detection system no longer utilizes this code due to the implementation of Neural Networks, the light adaptation code remains in use for hole segmentation.

This program was developed because lighting and reflections have a significant impact on image variability, making fixed threshold values unreliable for consistent object or hole detection. Therefore, a dynamic mask adaptation system was developed in order to adapt the thresholds of the code to the momentary lightning conditions. This means, that before beginning with any operation or starting any code, this must be first started by the user in order to adapt the thresholds. This allows a higher flexibility on the code, but it also has a downside. Each time the lightning conditions on the workspace change, the streaming loop will start to fail. So, when this happens, this code will be needed to be launched again and then the stream relaunched.

The program begins by importing all the libraries and initializing global variables. It then sets up the camera for both colour and depth streaming, discarding the initial 20 frames as on the stream. Afterwards, a set of frames is captured, and the colour image is cropped to a specific ROI to focus just o the object placing area. The cropped image is then displayed and used on the following steps to show the detected objects on it.

The core of the program is then run by the “adjust_threshold_parameters” function. This function creates a *Tkinter* window, allowing the user to select the colour space (RGB or HSV) used for thresholding. It then opens two more *OpenCV* windows called "Lower Thresholds" and "Upper Thresholds". Both windows work the same way, but one of them adjusts the upper limits of the threshold and the other one the lower limits. These windows are composed of 3 trackbars each, that allow the user to visually adjust the threshold values. In a continuous loop, the program reads the current positions of the trackbar and creates the onject detection threshold with them.

This threshold is then passed to the “Object_identification” function, which creates a binary mask based on the cropped RGB image. The function takes the cropped image and the threshold parameters, converts the image to the selected colour space and applies a thresholding limit to create a binary mask. It then finds the contours of the mask to detect object boundaries, filters these contours by area to remove noise, and draws the remaining ones on the RGB image. Then both the raw mask and an overlay of detected contours on the cropped image are displayed to the user, providing visual feedback on the effectiveness of the chosen values.

This loop continues until the ESC key is pressed. At this point, the save_thresholds function is called. This function converts the chosen thresholds into a Python list and then adds the selected colour mode to a JSON list named "thresholds.json". Following this, the program automatically calculates the values for hole thresholds based on these saved object thresholds. This dependency will always be constant because the changes in ambient light that affect the object will affect the holes similarly. As this is the case, a consistent relationship will always be maintained and it makes unnecessary to create 2 separate codes to determine both thresholds, making the process faster and more efficient. From both options, the object mask was chosen as the one to adjust because its detection is independent on the robot, making it more resilient and faster. Finally, this derived hole threshold is also saved, but in a separate JSON file named "thresholds_Hole.json."

4. Old object detection

The project's initial object detection system relied on a manual mask segmentation approach. This method, was adaptable to changing lighting conditions through the light adaptation code mentioned above, using them to define masks to segment objects. This system was effective for clearly separated, planar objects. However, it proved inefficient and unreliable when dealing with superimposed objects that lacked distinct boundaries. Consequently, this entire system, including its associated manual segmentation code, has been replaced by the Neural Network-based object detection module discussed in the main report.

This program's core functionality involves identifying and characterizing objects within an image using classical colour-based segmentation and contour analysis. From this segmentation it then calculates the meaningful information for the robot such as the size, orientation, and relative position relative to the robot of each detected object

The process begins with the “HomogeneousBgDetector” class. This class handles the initial detection of object contours, using the previously defined thresholds. It first crops the raw image to a specific ROI, to adjust the detection focus to the relevant area. It then loads the threshold values from the "thresholds.json" file, which was previously saved during the light adaptation process. Based on the specified colour mode specified on that JSON file, the cropped image is converted to it. A binary mask is then created, detecting pixels within the defined range. In this mask, the pixels within the threshold appear as white and are marked as object, whereas the others appear as black and are labelled as background. Afterwards, a contour-finding algorithm is applied to the mask, so that the object contours are identified. Finally, these contours are filtered by area to remove noise. The class returns these filtered contours and the generated mask.

Subsequently, the “identify_objects” function takes these detected contours as well as other camera data for further processing. So, after smoothing the detected contours for noise reduction, the function iterates through each object for feature extraction. For each object, it calculates the minimum-area bounding rectangle, providing its centre coordinates, width, height, and angle. This information is really useful and will help determining all the information demanded by the robot.

But the position parameters, along many others, tend to be really noisy sometimes. So, in order to reduce that noise and use more reliable information, the code uses global lists to collect multiple measurements over time for the same object and just use the median as explained on the Section 8 (Main problems). This approach is crucial for reducing noise in camera readings, as individual frames can contain inaccuracies. By taking the median of several measurements, transient errors are avoided, leading to more stable and reliable estimations of object parameters.

Depth measurements are then taken for each of the four bounding box corners. Sometimes, from all the measured points some tend to be invalid, as the depth frame has various “holes” specially on the edges. Therefore these invalid depth values are replaced with the average of valid depths from the same object, and the median depth for each edge is taken to further reduce noise. These 2D pixel coordinates and their median depths are then transformed into 3D coordinates relative to the camera. Here it is important to note that a fixed pixel-to-cm ratio cannot be used, unlike in other codes such as, hole detection. This is because objects can be placed at various heights within the workspace. An object closer to the camera will occupy more pixels, while the same object further away will occupy fewer. Therefore, using directl pixel ratios

would lead to inaccurate size measurements. Instead, the system uses the depth information from the camera to deproject the 2D pixel coordinates into 3D the space, providing true physical dimensions regardless of the object's position.

The Euclidean distances between pairs of the transformed 3D corner points are then calculated to determine the object's width and height in the 3D space. From both sides the larger dimension is always considered as height and the shortest as width. Based on these calculated width and height values, each object is classified as "Base," "Leg," "Finger," or "Noise object." At this point of the program the object's angle is also calculated, defining this as the longer side's angle.

Moreover, the object's depth at its centre are obtained from the bounding box as well. This 2D pixel point is converted into 3D coordinates relative to the camera, and a Transformation Matrix then converts these frames of reference to the robot's frame of reference. The angle is also adjusted to align with the robot's coordinate system, but this time using a simpler operation.

Nevertheless, if the object is a “Base” or a “Leg”, the holding point is stablished on the same centre. But if the object is a "Leg", a special adjustment is made to determine its holding point. To do so, the “find_nearest_intersection” function is used, which is the same as the one explained on the Section 7.3.4.

Finally, all collected robot coordinates and angles for each object are stored in global lists, in order to take their median values. The cropped image is then updated with the object's holding point, dimensions and angle, so that it can be displayed for the user. Lastly, the information for each detected object (type, position, size, angle) is appended to a list and saved to a JSON file named "Object_list.json", so that it can later be sent to the robot.

5. 3D point cloud

The capability for 3D point cloud generation of the workspace was developed with the initial idea of using it to develop 3D Neural Networks. The intention was to use this detailed 3D information for more robust environment understanding. However, the development of a 3D NN was ultimately not pursued due to incompatibilities between the development computer and essential libraries. Despite this, the 3D point cloud generation remains in use for visualization purposes, as it provides a valuable visual representation of the workspace, helping the user better understand the spatial context of the workspace.

The “generate_point_cloud” function is responsible for converting 2D colour and depth images into a 3D point cloud that can be visualized using the *Open3D* library just by using the camera’s intrinsic parameters. First, the depth image is converted to float32 and 2 empty lists are initialized to store the calculated 3D coordinates and their corresponding normalized RGB colour values.

Once this is created, the function initializes a loop that iterates through every pixel in the depth image. Here, depth value of each pixel is obtained. While doing so, a depth is also checked. If that value is 0, it indicates an invalid or unknown depth and therefore such pixels are ignored, preventing erroneous points from being added to the point cloud. Once the valid depth values are obtained, all the pixels are deprojected to the 3D space relative to the camera's optical centre. This calculated 3D points are then added to a points list. Simultaneously, the RGB colour for the same pixel is taken from the RGB image and normalized before

adding it to the homologous colour list. This normalization is done because the *Open3D* library expects the colour values to be between 0.0 and 1.0.

After processing all the pixels, the point and colour lists are converted into NumPy arrays and assigned to a previously created "o3d.geometry.PointCloud" object. This object is the final point-cloud, which can later be used for processing and visualizing.

An important remark to consider is that point cloud generation significantly slows down the streaming rate from approximately *6 FPS* to about *0.2 FPS*. Knowing this, and as is not strictly necessary for any part of the process to have the point-cloud, its use is avoided, preferring to work with RGB and depth frames independently. Therefore, this functionality is primarily used as a visual aid and is only shown when the user specifically demands it. This is done to avoid reducing the frame rate of the whole program, using it just when 3D information is needed by the user for a better spatial understanding of the workspace. All in all, even though the Open3D library itself could be very useful for the project (as it contains many potentially beneficial internal functions for various applications within this project), the point-cloud generation remains generally unused and the fast response of the code is prioritized.

Annex C: SDG impact report

On this annex, a final analysis of the project's impact on the SDG has been made. This analysis was made using *SDGtool [1],* an online tool that analyses each of the 17 goals profoundly, making a deep inspection of which aspects of each goal have the project worked on and which has been its impact for the planet in sustainability terms.

So, the following pages show an automatic generated report by this tool, explaining one by one all the goals that have been developed during the project and the impact had by it in each of them. In some of them the influence was direct with a mayor influence, while others were only minorly tackled. Nevertheless, all of them are summarized on the following Image 61.

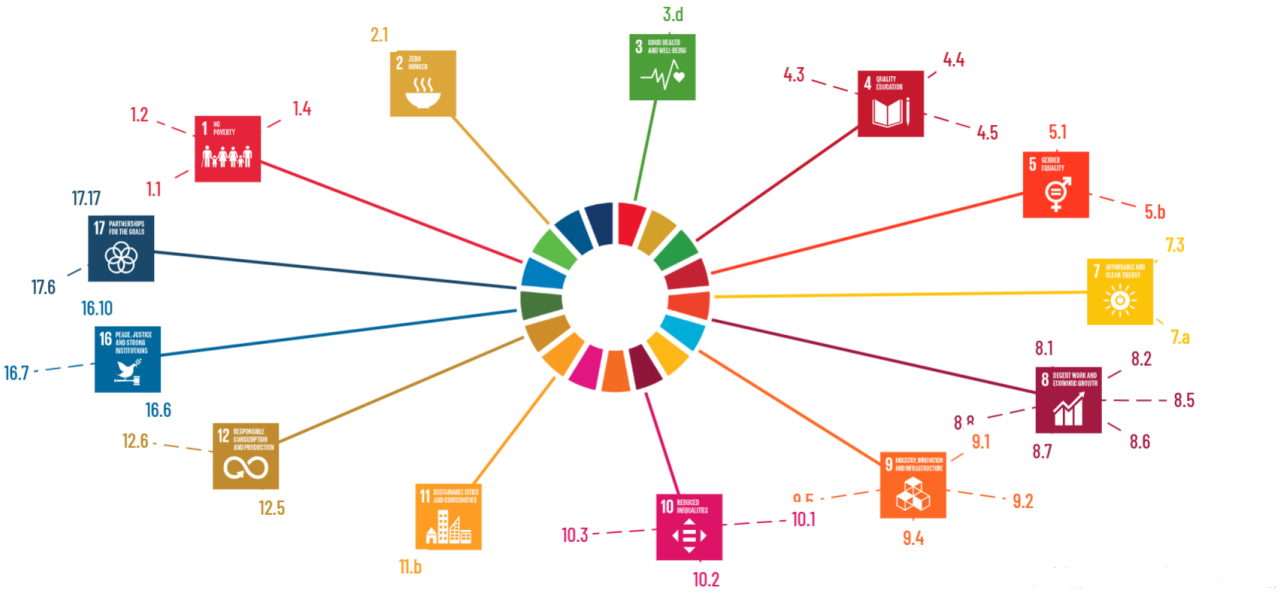


Image 61: SDG tool analysis result *[1]*

Table 22: SDG impact analysis via SDGtool *[1]*

End poverty in all its forms everywhere

By 2030**, eradicate extreme poverty** for all people everywhere, currently measured as people living on less than $1.25 a day

By 2030, reduce at least by half the proportion of men, women and children of all ages living in **poverty in all its dimensions according to national definitions**

By 2030 ensure that all men and women, particularly the poor and the vulnerable, have **equal rights to economic resources**, as well as access to basic services, ownership, and control over land and other forms of property, inheritance, natural resources, appropriate new technology, and financial services including microfinance

End hunger, achieve food security and improved nutrition, and promote sustainable agriculture

By 2030, **end hunge**r and ensure access by all people, in particular the poor and people in vulnerable situations including infants, to safe, nutritious and sufficient food all year round

---

Ensure healthy lives and promote well-being for all at all ages

Strengthen the capacity of all countries, particularly developing countries, for **early warning, risk reduction, and management of national and global health risks**

Ensure inclusive and equitable quality education and promote life-long learning opportunities for all

By 2030, ensure equal access for all women and men to affordable quality **technical, vocational and tertiary education, including university**

By 2030, substantially increase the number of youth and adults who have relevant skills, including technical and vocational **skills, for employment, decent jobs and entrepreneurship**

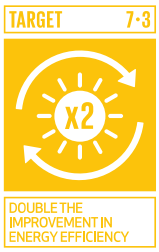
By 2030, eliminate **gender disparities in education** and ensure equal access to all levels of education and vocational training for the vulnerable, including **persons with disabilities, indigenous peoples and children in vulnerable situations**

Achieve gender equality and empower all women and girls

End all forms of **discrimination against all women and girls** everywhere

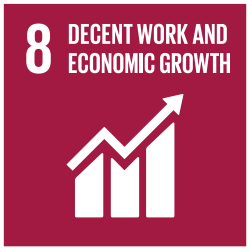
Enhance the use of **enabling technologies**, in particular information and communications technology, to promote the empowerment of women

Ensure access to affordable, reliable, sustainable, and modern energy for all



By 2030, double the global rate of improvement in **energy efficiency**

By 2030, enhance **international cooperation to facilitate access to clean energy research and technologies**, including renewable energy, energy efficiency, and advanced and cleaner fossil-fuel technology, and promote investment in energy infrastructure and clean energy technology



Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

Sustain per capita **economic growth** in accordance with national circumstances and, in particular, at least 7 per cent gross domestic product growth per annum in the least developed countries

Achieve higher levels of **economic productivity** through diversification, technological upgrading and innovation, including through a focus on high-value added and labour-intensive sectors

By 2030, achieve full and **productive employment and decent work** for all women and men, including for young people and persons with disabilities, and equal pay for work of equal value

By 2020, substantially reduce the proportion of **youth not in employment, education or training**

Take immediate and effective measures to eradicate **forced labour**, end modern slavery and human trafficking and secure the prohibition and elimination of the worst forms of child labour, including recruitment and use of child soldiers, and by 2025 end child labour in all its forms

Protect **labour rights** and promote safe and secure working environments of all workers, including migrant workers, particularly women migrants, and those in precarious employment

Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

Develop quality, reliable, sustainable and resilient **infrastructure**, including regional and transborder infrastructure, to support economic development and human well-being, with a focus on affordable and equitable access for all

Promote inclusive and sustainable **industrialization** and, by 2030, significantly raise industry's share of employment and GDP in line with national circumstances, and double its share in LDCs

By 2030, upgrade infrastructure and **retrofit industries to make them sustainable**, with increased resource-use efficiency and greater adoption of clean and environmentally sound technologies and industrial processes, with all countries taking action in accordance with their respective capabilities

Enhance scientific research, upgrade the technological capabilities of industrial sectors in all countries, particularly developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers per 1 million people and public and private **R&D spending**

Reduce inequality within and among countries

By 2030, progressively achieve and sustain **income growth of the bottom 40 per cent** of the population at a rate higher than the national average

By 2030, empower and promote the social, **economic and political inclusion of all** irrespective of age, sex, disability, race, ethnicity, origin, religion or economic or other status

Ensure **equal opportunity** and reduce inequalities of outcome, including through eliminating discriminatory laws, policies and practices and promoting appropriate legislation, policies and actions in this regard

Make cities and human settlements inclusive, safe, resilient and sustainable

By 2020, substantially increase the number of cities and human settlements adopting and implementing integrated **policies and plans towards inclusion, resource efficiency, mitigation and adaptation to climate change, resilience to disasters**, develop and implement in line with the Sendai Framework for Disaster Risk Reduction 2015-2030, holistic disaster risk management at all levels

---

Ensure sustainable consumption and production patterns

By 2030, substantially reduce **waste generation** through prevention, reduction, recycling, and reuse"

Encourage **companies**, especially large and trans-national companies, **to adopt sustainable practices** and to integrate sustainability information into their reporting cycle

Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels

Develop **effective, accountable and transparent institutions** at all levels

Ensure responsive, **inclusive, participatory and representative decision-making** at all levels

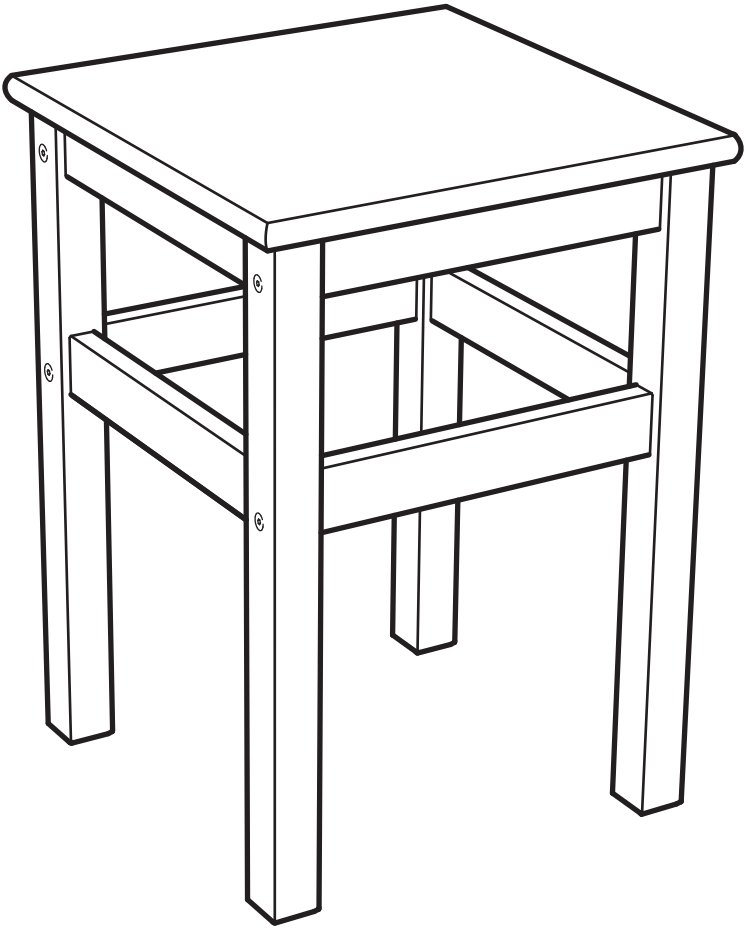
Ensure public **access to information** and protect fundamental freedoms, in accordance with national legislation and international agreements

Strengthen the means of implementation and revitalize the global partnership for sustainable development

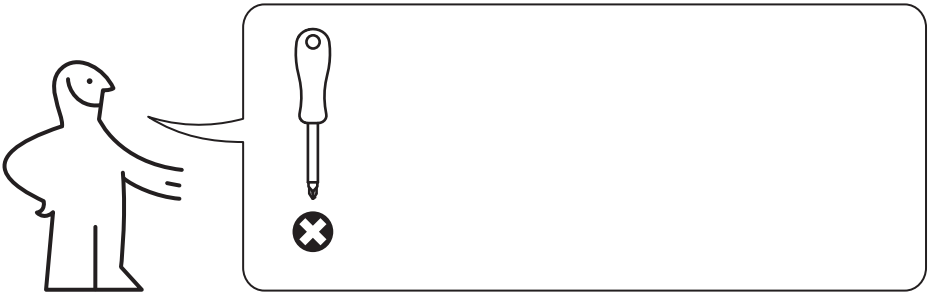
Enhance North-South, South-South and triangular regional and international cooperation on and **access to science, technology and innovation, and enhance knowledge sharing** on mutually agreed terms, including through improved coordination among existing mechanisms, particularly at the UN level, and through a global technology facilitation mechanism

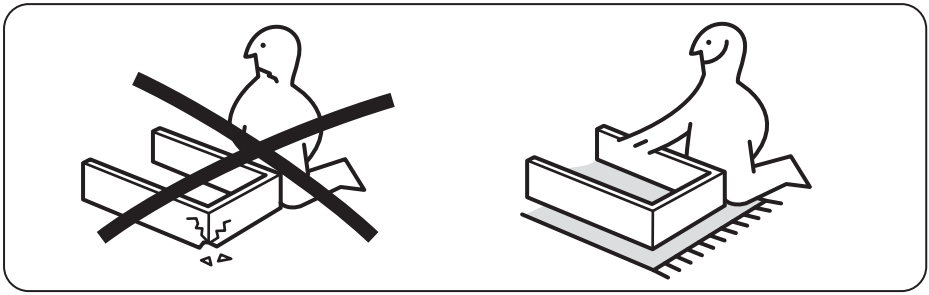
Encourage and promote effective **public, public-private, and civil society partnerships**, building on the experience and resourcing strategies of partnerships

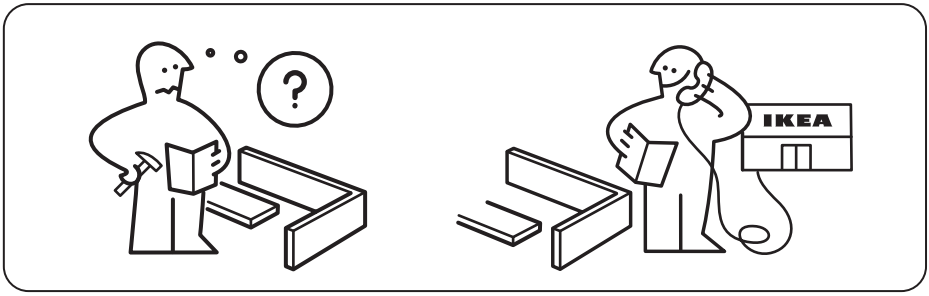
ODDVAR

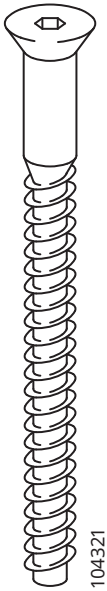


Design and Quality
IKEA of Sweden

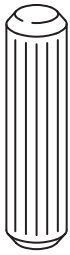





?
IKEA

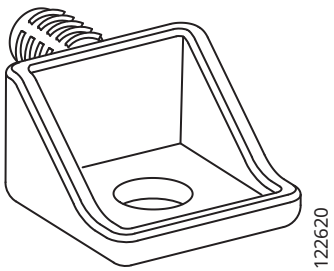


8x

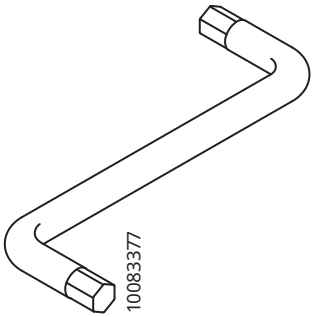


8x

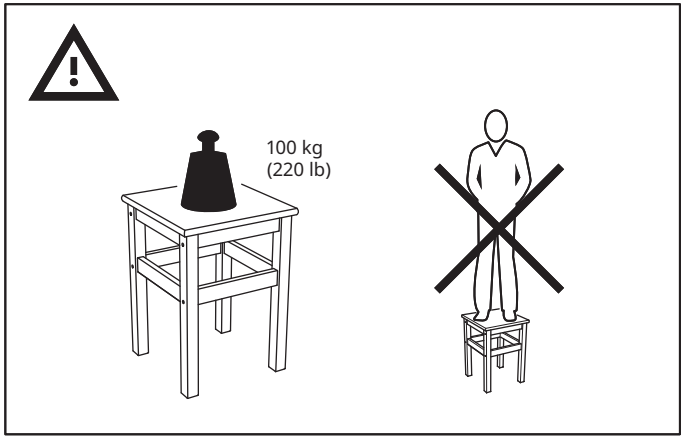
4x



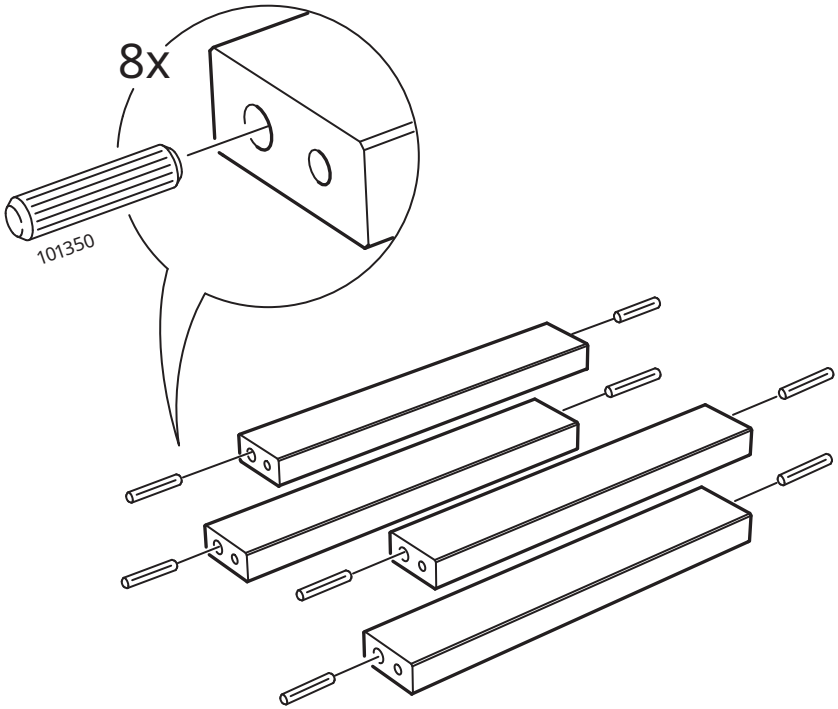
4x



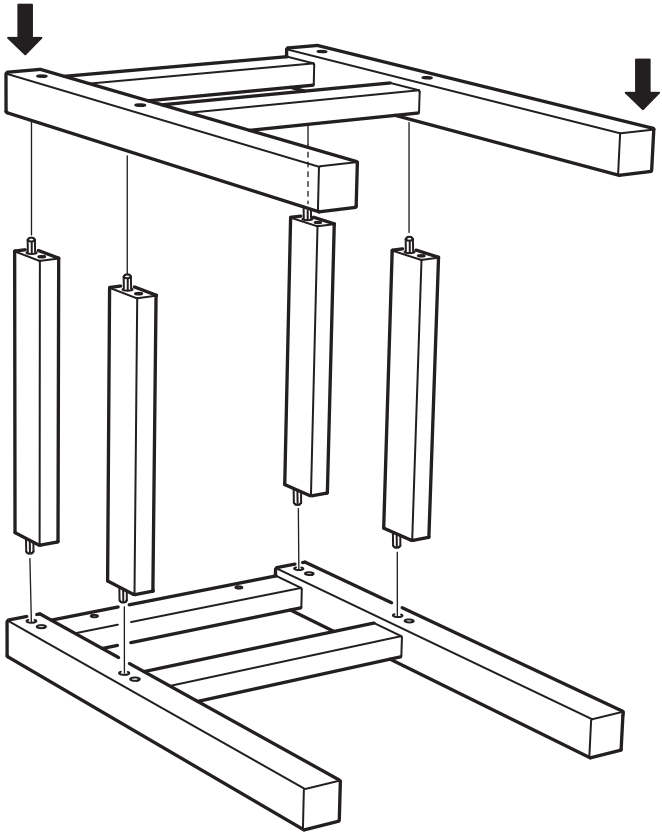
1x



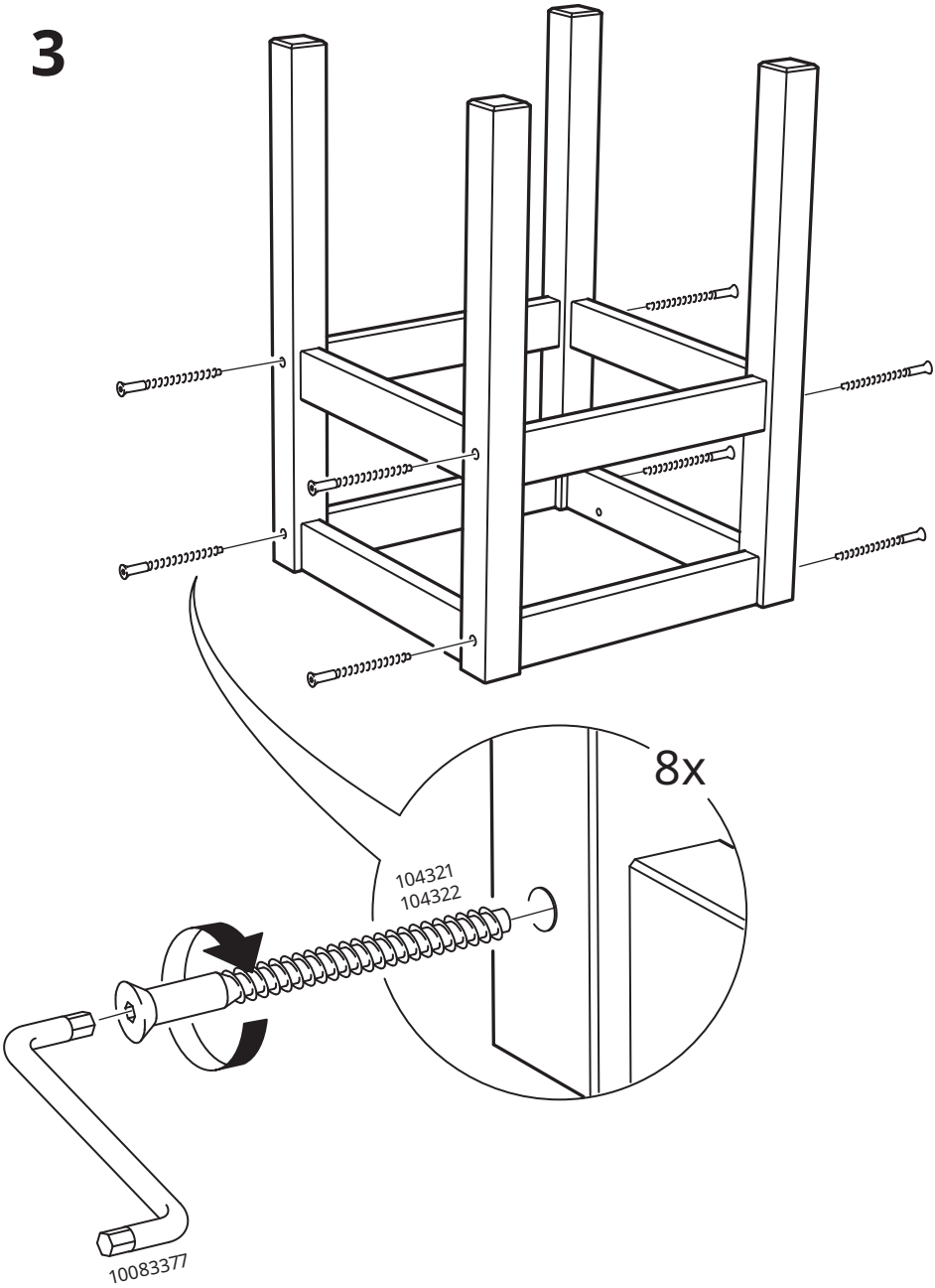
1



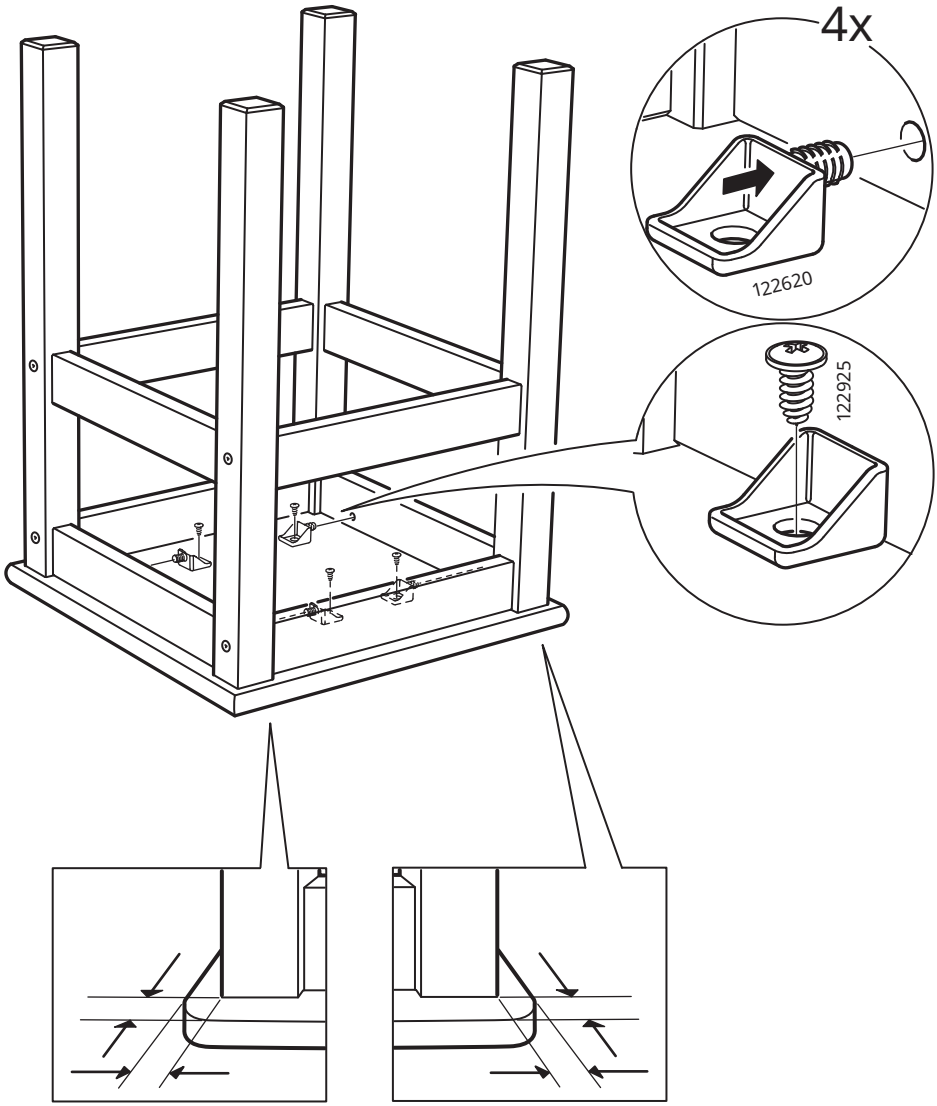
2



3



4



2021-03-25 AA-766928-4

Imprimir formulario Limpia formulario

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

NORMAS DE ESTILO PARA LA PRESENTACIÓN DEL TFG Y DECLARACION DE HONESTIDAD ACADEMICA

La tipografía (tipo de letra/tamaño) y maquetación (sangrías, párrafos, listas o viñetas…) tienen un formato libre pero debe permitir una fácil lectura del documento y el alumno debe respetar ese formato en TODO el documento.

Marque las casillas que cumple y adjunte esta hoja a la entrega del documento (debe cumplir todos los puntos para ser presentado):

☑ Conforme al código de honestidad académica, certifico que el trabajo presentado es original y desarrollado íntegramente por mí, citando adecuadamente las fuentes externas que haya utilizado en la realización de este trabajo.

☑ Las páginas están numeradas en la parte inferior derecha (numeración correlativa de todos los capítulos).

☑ Contiene índice, asociando los temas a la numeración de página.

☑ Si se utilizan tablas, figuras o ecuaciones, siguen una numeración correlativa. Una serie para tabla, otra para figuras y otra para ecuaciones. (El alumno puede elegir si el título de estos elementos está encima o debajo del elemento, pero mantiene el mismo criterio en todos).

☑ Se cita correctamente las fuentes en el texto formato Autor (Año). Ejemplos: http://blog.apastyle.org/apastyle/2011/01/writing-in-text-citations-in-apa-style.html

Cita directa: los sistemas..... (Pérez y Martínez, 2007; Alba, 2010)
Cita indirecta: como afirman Pérez y Martínez (2007) los sistemas.....
Cita con más de dos autores: (Gutiérrez y otros, 2003)

☑ Si se citan fuentes externas, existe sección de Bibliografía con las referencias formateadas adecuadamente (se sugiere formato APA: http://www.apastyle.org/). Ejemplos recomendados en:

http://www.upv.es/laboluz/master/metodologia/textos/citar.pdf
https://biblioguias.uam.es/citar/estilo_apa
http://www.ub.edu/biblio/citae-e.htm

☑ Si se citan fuentes externas, existe concordancia entre las citas en texto y la lista de referenciadas.

☑ Sin faltas de ortografía, ni errores tipográficos.

☑ Figuras e imágenes de buena calidad (si no existen figuras o imágenes, marca la casilla como cumplida).

☑ Si hay gráficas, estas son claras y están bien etiquetadas (si no existen gráficas, marca la casilla como cumplida).

☑ Si hay ecuaciones, estas son claras y están bien escritas (si no existen ecuaciones, marca la casilla como cumplida).

☑ Uso correcto de símbolos, anagramas, denominaciones, etc. (si no existen estos elementos, marca la casilla como cumplida).

Firmado por DAÑOBEITIA CAPETILLO KERMAN - ***9669** el día

Fecha: 25/06/2025 Firma:_____

Plantilla para volcar puntuaciones de las rúbricas

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

Título TFM: DESIGN, DEVELOPMENT, AND EVALUATION OF A FLEXIBLE MANUFACTURING CELL USING VISUAL GUIDANCE FOR COLLABORATIVE ROBOTS FOR AUTOMATIC ASSEMBLY AND MOUNTING OPERATIONS

Autor TFM: Kerman Dañobeitia Capetillo

Tutor TFM: Antonio José Sánchez Salmerón

Evaluador: EL ALUMNO

Use las descripciones detalladas de la rúbrica para seleccionar puntuación.
Marque con un tick la casilla correspondiente.

ELABORACION TFM	Exc	Alto	Med	Insuf	Defic
P-01 Viabilidad del proyecto y/o de su trabajo de campo	■				
P-02 El alumno sabe recoger la información necesaria	■				
P-03 Comprensión de la tarea	■				
P-04 Motivación, iniciativa e independencia	■				
P-05 Organización y planificación de las tareas		■			
P-06 Innovación, creatividad y emprendimiento	■				

COMENTARIOS:

INFORME TFM depositado	Exc	Alto	Med	Insuf	Defic
I-01 Comunicación escrita	■				
I-02 Introducción-Objetivos	■				
I-03 Antecedentes	■				
I-04 Metodología/ Desarrollo	■				
I-05 Resultados		■			
I-06 Conclusiones	■				
I-07 Documentaci-ción	■				
I-08 Honestidad académica	■				

COMENTARIOS:

Firma:

Firmado por DAÑOBEITIA CAPETILLO KERMAN - ***9669** el día

Plantilla para volcar puntuaciones de las rúbricas

Título TFM: DESIGN, DEVELOPMENT, AND EVALUATION OF A FLEXIBLE MANUFACTURING CELL USING VISUAL GUIDANCE FOR COLLABORATIVE ROBOTS FOR AUTOMATIC ASSEMBLY AND MOUNTING OPERATIONS

Autor TFM: Kerman Dañobeitia Capetillo

Tutor TFM: Antonio José Sánchez Salmerón

Evaluador: El TUTOR

Use las descripciones detalladas de la rúbrica para seleccionar puntuación.
Marque con un tick la casilla correspondiente.

ELABORACION TFM	Exc	Alto	Med	Insuf	Defic
P-01 Viabilidad del proyecto y/o de su trabajo de campo	■				
P-02 El alumno sabe recoger la información necesaria	■				
P-03 Comprensión de la tarea	■				
P-04 Motivación, iniciativa e independencia	■				
P-05 Organización y planificación de las tareas	■				
P-06 Innovación, creatividad y emprendimiento	■				

COMENTARIOS:

INFORME TFM depositado	Exc	Alto	Med	Insuf	Defic
I-01 Comunicación escrita	■				
I-02 Introducción-Objetivos	■				
I-03 Antecedentes	■				
I-04 Metodología/ Desarrollo	■				
I-05 Resultados	■				
I-06 Conclusiones	■				
I-07 Documentaci-ción	■				
I-08 Honestidad académica	■				

COMENTARIOS:

Firma:

Firmado por ANTONIO JOSE SANCHEZ SALMERON - NIF:***8627** el día 25/06/2025 con un certificado emitido por ACCVCA-120