POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Framework for Safe Mobile Manipulation in Human-Centered Applications

Original
A Framework for Safe Mobile Manipulation in Human-Centered Applications / Cen Cheng, P.D., Blengini, C.L., Cavelli, R.F., Ripi, A., Indri, M.. - In: ROBOTICS. - ISSN 2218-6581. - ELETTRONICO. - 15:4(2026). [10.3390/robotics15040068]

Availability:
This version is available at: 11583/3009690 since: 2026-04-08T08:43:45Z

Publisher:
MDPI

Published
DOI:10.3390/robotics15040068

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

19 November 2026

robotics

MDPI

Review

A Framework for Safe Mobile Manipulation in Human-Centered Applications

Pangcheng David Cen Cheng *, Cesare Luigi Blengini, Rosario Francesco Cavelli, Angela Ripi and Marina Indri *

Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; cesare.blengini@polito.it (C.L.B.); rosario.cavelli@polito.it (R.F.C.); angela.ripi@studenti.polito.it (A.R.)
* Correspondence: pangcheng.cencheng@polito.it (P.D.C.C.); marina.indri@polito.it (M.I.)

Abstract

In recent years, applications with robots collaborating actively with humans have been increasing. The transition from Industry 4.0 to 5.0 rearranges the focus of fully automated processes to a human-centered system that allows more customization and flexibility. In human-centered systems, the robot is expected to safely assist or provide support to the human operator, avoiding any unintentional harm, while the latter is focused on tasks that require human reasoning, since current decision-making systems still have some limitations. This survey reviews all the main functionalities required to make a robot (collaborative or not) act as an assistant for human operators, analyzing and comparing solutions proposed by the authors (based on previous works) and/or the ones available in the literature. In this way, it is possible to combine those functionalities and build a complete framework enabling safe mobile manipulation while interacting with humans. In particular, a mobile manipulator is used to receive requests from a user, navigate in a human-shared environment, identify the requested object, and grasp and safely deliver such an object to the user. The framework, which is completed by a user interface designed using Android Studio, is developed in ROS1, tested, and validated on a real mobile manipulator in real-world conditions.

Keywords: human–robot collaboration; mobile manipulation; safe human–robot interaction

1. Introduction

In recent years, robotics applications have been moving towards more collaborative interaction with humans. This trend can also be noted in the evolution of the use of techniques based on Artificial Intelligence (AI) to improve the productivity of human tasks in several contexts.

While in Industry 4.0 scenarios, all the processes are automated, often avoiding any human intervention, the recent Industry 5.0 paradigm aims at putting humans at the center of the production line, assigning to the robotic systems the task of assisting the human operator [1]. This approach stems from the fact that there are many tasks that cannot be automated with the precision and speed that a human operator can provide, since they benefit from the worker's experience. For instance, among the operations that still require the expertise of a human operator [2], there are the assembly of custom pieces [3] and disassembly operations (e.g., battery management and cloth recycling). This context can also be extended to other scenarios as well, where a human needs some support or assistance, e.g., for the search and retrieval of specific objects that are necessary to perform a task.

Check for updates

Academic Editors: Stefano Baraldo and Xinjun Liu

Received: 31 January 2026
Revised: 23 March 2026
Accepted: 24 March 2026
Published: 25 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Human–robot interaction (HRI) applications have been investigated extensively in the last decade, based on collaborative robots that can safely work in the vicinity of human workers. The possible interaction forms can be classified as coexistence, cooperation, and collaboration. The differences are related to the existence or not of a common goal and to the specific tasks of humans and robots. Coexistence simply means that the working space is shared; cooperation is a matter of performing independent tasks converging toward a common goal, while collaboration usually implies direct contact or interaction to achieve the same goals.

While safety is crucial to ensure a smooth interaction between humans and robots, in a workplace, most of the cases where humans get hurt occur when they are distracted [4]. However, this type of risk can be reduced when human operators are used to working with robots, also from a psychological point of view [5], and the robotic system is programmed to avoid any unintentional collision with people.

Most of the existing frameworks that include the human as the core element in robotic applications are built using collaborative robots (cobots) with a set of sensors as additional safety measures for human operators. The setup of the cobots is intrinsically safe, since they are designed to work closely with humans and are compliant with the safety standards, such as ISO 10218 and ISO/TS 15066 [6]. It is worth noting that the cobots used in such frameworks are fixed-base [7], which means that they have limited working space and cannot fully assist the human operator in any case. Mobile manipulators in this context can be preferable; some recent updates related to ISO 10218:2025 have consolidated the collaborative safety within the main standard.

This paper aims at presenting a framework that enables safe collaboration and interaction between humans and robots. There is much research on this kind of topic; however, the expensive robotic systems that are often employed might not be affordable for many small and medium enterprises, which could leverage robotic systems to achieve greater customization than mass production. In addition, the algorithms used to ensure safe interaction are computationally expensive, since they need a lot of resources, data, and time. The context of this work was initially introduced in [8] and aims to provide a flexible software architecture with additional safety layers that allow the robot to collaborate with the human operator in assistive tasks. In such a framework, some key functionalities are identified, such as AI-based recognition for the humans and robots' working spaces, safe path planning, and safe interaction strategies for both humans and robots when their workspaces intersect. To the best of the authors' knowledge, most of these functionalities are available in the literature, but implemented and validated only separately, without the complete integration of all the features in a single framework to enable safe HRI.

The framework here proposed is not only limited to applications that foster Industry 5.0 standards, but also can be applied in other contexts where humans still need support or assistance from a robot, and the robot carries out all the necessary tasks without the risk of unintentionally hurting the user. The main motivation is that, in a human-centered workspace, humans should be concentrated on performing their tasks so as to improve their performance and reduce the risks due to distractions. In order to ensure such working conditions, the robots should ideally be able to carry out all the side tasks, understanding human gestures while addressing proxemics criteria.

The main contribution of this paper relies on reviewing existing solutions as key functionalities for the development of a framework for low-resource mobile manipulators, while also limiting the use of external sensors, so as to provide the community with a versatile base framework that can be easily adapted and enriched to other kinds of robotic systems. It must be noted that the proposed approach is an evolution of previous works, where several functionalities were developed and tested separately; here, the most

significant results are combined with an intuitive user interface to provide a complete baseline for a safe robot-to-human handover in a space where both elements coexist. Hence, such functionalities are compared with existing solutions from the literature, thus providing an overview of the base features that a framework for safe HRI should include.

The rest of this paper is structured as follows: Section 2 presents an overview of related works. Section 3 describes the proposed framework, where the main functionalities are compared with previous authors' work and other solutions from the literature. Section 4 illustrates an experimental setup to test the components of the whole framework within an overall case study. The performances achieved in the experimental validation are discussed in Section 5. Section 6 concludes this paper, addressing current limitations and future research directions.

2. Related Works

In the literature, as far as safety for HRI is concerned, there are strategies that should be adopted depending on whether a collision occurs or not. Usually, these strategies are referred to as pre-collision or post-collision [9]. Pre-collision strategies are those used to avoid at any cost the collision of the robot with other agents and humans. This can be done by properly setting safe distances, using safe velocity constraints depending on the vicinity of the human to the robot workspace, and applying safe path and handover planning procedures. On the other hand, post-collision approaches try to mitigate the damage in the case that unintentional harm is done to the human [10]. By combining both strategies in an HRI application, the human should be protected from any unintentional harm, or in the case that harm occurs, it should be mitigated. However, in order to prioritize human safety in a shared workspace, pre-collision strategies should be adopted [11]. Even though including pre- or post-collision strategies in any human–robot collaborative framework is important, it is not enough to guarantee a comfortable interaction between humans and robots, since most humans might not be used to having autonomous systems moving very close to them, especially if such systems carry objects to be delivered to several places.

Another aspect that should be considered for a safe and comfortable interaction with robots is communication; good verbal or non-verbal communication enhances the user's trust towards elements like robots, since they are perceived as partners [12,13]. Cues and safety distances for HRI may vary depending on the robotic system, since different criteria should be considered for manipulators or mobile robots, but in the case of mobile manipulators, these criteria can be combined. In the literature, there are many studies regarding verbal and non-verbal forms of communication between humans and robots, trying to mimic how humans interact. Usually, verbal communication can be applied only in a limited context for HRI, and it is preferred for social affective interactions with robots [14]. On the other hand, non-verbal communication between robots and humans in HRI is preferable in general, since it is more versatile and is not bounded by personal experiences or culture [15,16]. Furthermore, non-verbal forms of communication, e.g., based on proxemics, haptics, visual gaze, hand gestures, and body postures, are consistent and socially accepted [17]. On top of that, learning some policies or introducing communication cues in an HRI context could improve the interaction related to manipulation tasks like handovers.

The authors in [18] presented a collision-free path planning algorithm for manipulators in a safe HRI scenario. In the case that the human (modeled as capsules) is close enough to the manipulator, it triggers a replanning optimization problem to avoid the human while reaching the target pose. Safe distances are computed with the Gilbert–Johnson–Keerthi algorithm, but the approach is tested only in simulation, with a limitation related to the orientation of the tip, which is fixed. Moreover, according to the work in [19], safety in Human–Robot Collaboration (HRC) scenarios is improved by the context awareness,

thanks to the capability of recognizing human poses. Nevertheless, the human recognition of the collision sensing module requires three RGB-D cameras for the robot hand–eye calibration setup. In [20], a framework is proposed for a manipulator to learn some policies for robot-to-human handover and receive feedback about the human preferences when objects are handed over, but this approach does not consider the human response during the object handover.

In [21], haptic cues are used as a communication channel for human-to-robot, robot-to-human, and even robot-to-robot handover control, thanks to the force sensors located in the fingertips. Moreover, in [22], safe multi-channel communication is proposed to enhance HRC. The interface is called DiGeTac, a gesture-sensing module that recognizes hand signs to enable complex collaborative tasks like assembly/disassembly, handover, etc. The work presented in [23] investigates human–human handover mechanisms and adapts them to human–robot handovers. In particular, it considers the eye gaze as the feedback signal to transfer the object. After many tests with different people, the work suggests implementing an HRI framework with a sequence of face–hand–face gaze transitions to ensure a more effective object handover.

On the other hand, the safety criteria for mobile robots include additional layers to classical path planning algorithms, which are aimed at computing a collision-free path while considering a purely static environment; in the literature, they are usually referred to as global planners. In order to deal with dynamic obstacles, a local planner is included in the planning structure to modify the original path based on the real-time readings from the robot's sensors in order to avoid any moving obstacles. Some well-known local planners in ROSs (Robot Operating Systems) are the Dynamic Window Approach (DWA) and Timed Elastic Band (TEB); however, their performance is limited, since they do not consider the obstacle's motion information, but only static obstacles at every time step [24]. In this regard, the combination of global and local planners is not sufficient to ensure the safety of humans, since their motions are often unpredictable. In order to overcome this problem, one might include a social navigation layer; for instance, the authors in [25] modified the TEB into the GTEB (goal-oriented TEB) to be reactive to humans while computing a socially feasible trajectory. Human–robot proxemics are considered in other studies, such as in [26], where the cost of the person's space is modeled by a Gaussian-like distribution, proportional to the person's direction of movement.

Finally, other approaches include reinforcement learning with policies related to proxemic-informed rewards in order to enhance social comfort [27]. It is worth noting that human-friendly and socially acceptable robotic behavior enhances the safety perception of the human towards the robot [28].

3. Framework Description

The framework considered here is an evolution of previous works, with the aim of describing and developing the necessary tools for safe interactions between humans and robots. The considered scenario includes mobile manipulators performing some operations in an environment shared with other agents and humans; on demand, one of the mobile manipulators can be requested to retrieve and deliver specific items. Depending on the application, the items might be simple objects to be delivered to the user (cups, spoons, bottles, etc.) or tools (screwdrivers, scissors, blades, etc.) that a user needs to complete a task in his/her working station.

With this context in mind, it is possible to identify at least the following basic operations that the robot must perform: (i) object information processing, (ii) safe navigation, and (iii) robot-to-human handover. On top of that, providing an intuitive user interface to command the robot will close the gap between non-robotic experts and robots in many

applications. As a consequence, the development and adaptability of new activities can be executed faster.

A high-level overview of the proposed framework is shown in Figure 1, which is an evolution of the work presented in [8]. In particular, the object information processing block includes all the functions to: train the framework to identify specific objects, do object segmentation and compute its grasping pose, identify and track the human in the scene, and recognize hand gestures. The safe navigation function concerns the navigation layers for the mobile platform while also considering the human as a special obstacle, and the robot-to-human handover feature allows the robot to deliver a requested object while satisfying some safety constraints.

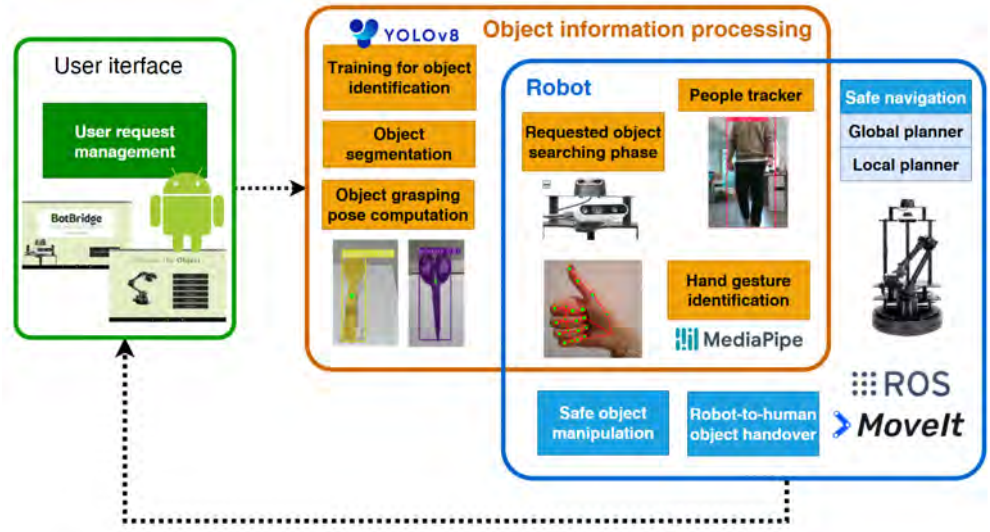


Figure 1. A high-level view of the framework components.

The framework's basic operations will be developed adopting pre-collision strategies and settings (as discussed in Section 2) for whatever concerns collision avoidance and policies for safe collaboration with humans. In particular, this choice implies that all the proposed algorithms will be implemented adopting proper safety distances for HRI; post-collision strategies will not be considered in this work. Even though there are several available algorithms in the literature, here we will consider those that can also be deployed on systems with low computational resources, e.g., limited CPU resources, limited sensors, no GPUs, etc. The robotic system considered here is a mobile manipulator to exploit its versatility to manipulate objects and move around.

3.1. Object Information Processing

There are many algorithms used for object manipulation, which include object identification, pose estimation, and motion planning to grasp/deliver such objects. Object identification algorithms often require image/video data to detect the desired item; the object's pose is then estimated to let the robot plan the trajectory to reach such a pose. One strategy is to use fiducial markers such as ARTags and ArUco, which can be attached to the objects, and estimate their pose exploiting the appropriate marker reader [29]. One of our previous works, presented in [30,31], employs ARTags to provide useful information about the object of interest, proposing a low-resource algorithm to recognize the object position and estimate its pose for grasping.

A strength of this choice is its easy implementation: using ARTag readers of specific objects reduces the computational burden and gives the possibility of dedicating most of the resources to the computation of the robot path planning and control, as well as to the

management of (static and dynamic) obstacles. The disadvantages are mainly related to the size of the objects and of the tags. On one hand, small tags can be difficult to detect; on the other hand, large tags on small objects might cause some difficulties during grasping.

Beyond fiducial markers, object detection and segmentation algorithms offer alternative strategies for identifying objects and planning safe grasps, each one with different trade-offs in terms of computational cost, accuracy, and safety awareness.

Bounding box detection with human-in-the-loop correction. Object detection algorithms such as YOLO [32] localize objects in the scene using labeled bounding boxes, enabling the robot to classify objects and support grasp planning. In [33,34], we integrated an object detection module based on YOLOv5s, allowing the robot to detect and manipulate objects of varying shapes and sizes. A simple approach consists of selecting the center of the bounding box as the grasp point for a parallel gripper, as in Figure 2. However, bounding boxes alone do not encode information about dangerous object parts. To address this, the operator can indicate the location of the object's dangerous region after detection. As illustrated in Figure 2, the red point corresponds to the center of the bounding box, while the operator specifies the hazardous region by clicking it directly on the image (blue point); the robot computes the grasp point (green point) as an intermediate location along the line connecting the two previous points. This allows the robot to grasp the object near the dangerous part, leaving the safe one free for human handover. The main advantages of this approach are simplicity and low computational cost, but the required user input may interrupt or distract the operator from his/her primary task.



Figure 2. Predicted grasping point (red), point indicated by the user (blue), and grasping point for the robot (green).

Segmentation-based grasp planning. Compared to bounding boxes, segmentation models provide a more precise outline of the object, capturing detailed shape information that can support more informed grasp planning. YOLO segmentation variants can be combined with object-agnostic grasp pose estimation methods, such as GraspNet [35], AnyGrasp [36], GraspGen [37], and Contact-GraspNet [38]. These approaches exploit object affordances to identify stable grasp configurations, and can generate candidate grasps more effectively, by isolating the target object from the scene using its segmentation mask. The main drawback is that their performance may depend on object geometry or size; in addition, they do not explicitly consider the safety of the handover phase, since no distinction is made between the graspable and the dangerous parts of the object.

Affordance keypoint estimation. To explicitly account for user safety during human–robot interaction, a more structured representation of object parts can be adopted. In [39,40], we proposed an approach that leverages part affordances to divide each object into three functional regions, as illustrated in Figure 3: (i) the handle, most suitable for human grasping; (ii) the dangerous part (e.g., blades, spikes, or tool tips); and (iii) the most suitable region for robot grasping, typically located between the first two. These

affordance regions can be represented as segmentation masks (Figure 4a) or more compactly as keypoints (Figure 4b), where each keypoint approximates the center of one affordance region. The object class, the bounding box, and the keypoints are predicted using a custom version of YOLOv8Pose trained on a dataset of cluttered scenes. This approach enables fully automatic grasp planning that is inherently aware of safety-relevant object features, without requiring any input from the operator; however, it requires annotated training data and offers limited flexibility for users who may prefer a different handover configuration.

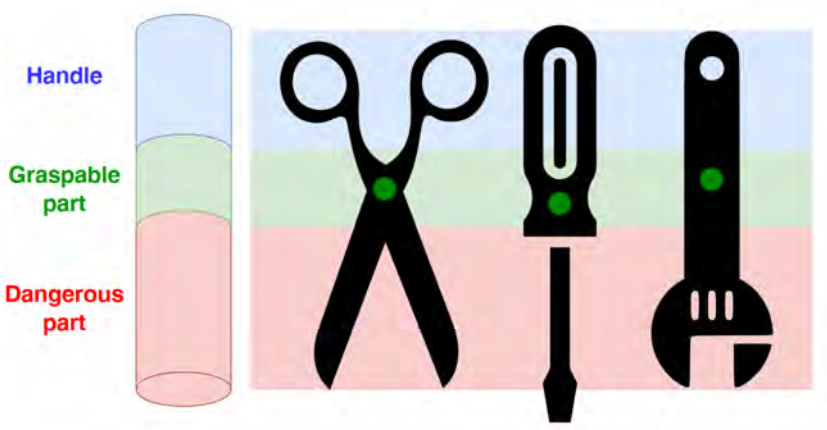


Figure 3. Object segmentation for the tools. Blue indicates the handle for humans, green corresponds to the graspable part for the manipulator, and red is the dangerous part (usually the tool) of the object. The green dot is a possible grasping point for the manipulator's path planner.

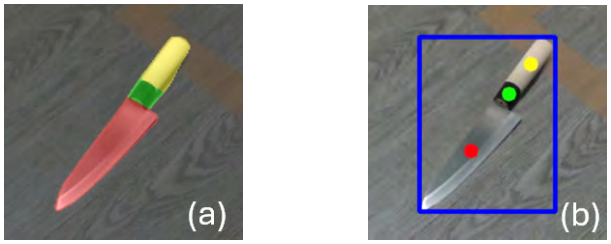


Figure 4. (**a**) Affordance mask and (**b**) affordance keypoints of a knife. Red, green, and yellow marks refer to *danger*, *grasp*, and *handle* affordances, respectively.

To sum up, Table 1 compares the algorithms tested and deployed by the authors with the ones available in the literature.

Table 1. Comparison of approaches to determine the object's grasping pose.

	Approach	Main Features	Main Limitations
Fiducial markers	ARTag * [30], ArUco [41]	Easy to read and easy to implement	Limited to the object size
Object detection algorithms	YOLOv5s * [33], YOLOv8 * [32,39]	Fast algorithm to identify objects	Bounding box is large, and image CoM is not always the object's CoM
Object semantic segmentation	YOLOv8n-seg [42]	Each pixel of the image can be labeled by a class	Struggles with multiple packed objects
Object instance segmentation	YOLOv8s-seg [43]	Similar to semantic segmentation, but each object has a unique identifier	Additional processing layers, slower than semantic segmentation
Affordance estimation	GraspNet [35], AnyGrasp [36], GraspGen [37], Contact-GraspNet [38]	Several grasping poses for the object are determined	Computationally expensive and struggles on some object shapes

(*) Approaches that have been tested by the authors in previous works.

3.2. Safe Navigation

The mobile manipulator is expected to move autonomously in an environment where other humans and robots coexist, so one of the most fundamental aspects that ensures the safety of humans is the fact that mobile robots must be able to avoid static obstacles, as well as dynamic ones, such as human operators. One strategy to deal with this kind of problem is to combine different path planning layers, i.e., a global planner and a local planner. The global planner takes the information from a static map and, under some optimization criteria, computes the best trajectory for the robot to reach a specific destination without considering changes in the environment. Among the available global path planners, there are: A*, Dijkstra, Probabilistic Roadmap (PRM), rapidly exploring random tree (RRT), artificial potential fields, barrier functions, etc.

On the other hand, local planners update the global plan to avoid obstacles (moving or not) that were not recorded while the map was being built. Dynamic obstacle avoidance is made possible using real-time data from the robot's sensors to deform the path around the obstacle, while guiding the robot towards its original destination. For instance, the DWA [44] and the TEB [45] are two of the most used algorithms that compute a feasible path to avoid moving obstacles. The choice of the combination of global–local planners depends on the application and its constraints, but common choices include the A* algorithm with the TEB [46] or DWA [47].

However, typical local planners cannot efficiently deal with human obstacles: they do not consider a safe distance when considering humans, so the robot might unintentionally hit them. In this case, a social navigation layer could be included in the hybrid global–local path planning structure, so the human is identified as a special obstacle, and the robot can react accordingly with additional safety measures [48]. Depending on the context and specific application requirements, humans can be modeled by considering their instantaneous positions, speed, intentions, or motion prediction [49]. In this regard, most of the sensors used to detect and track humans are LIDARs and RGB-D cameras [50], and their data can be used to train planners, such as imitation learning-based, human position-based, human prediction-based, safety-aware, etc., which can also be combined. For instance, the work in [51] presents a deep reinforcement learning approach to train mobile agents to decide which are the most suitable local planners for different scenarios.

In our previous work in [31,52], an obstacle avoidance strategy based on a varying costmap shape is presented. The approach allows the detection of humans moving near the robot and represents their costmaps by means of Gaussian-shaped areas, which are proportional to the humans' speed. This way, an additional safety layer is added to the local planner, enabling the robot to modify its path, while keeping a safe distance from the human and ensuring collision avoidance. The approach exploits the social navigation layer [53], which adds custom layers to the costmap package in the ROS. Note that this kind of approach implements proxemics rules, so that a safe distance between humans and robots is always considered for the path planning. In this way, instead of inflating the area around all the obstacles to avoid collision, the robot is able to distinguish humans and optimize its path, prioritizing the spaces that are not occupied by people. Figure 5 depicts the Gaussian area around the human while he/she is moving near the mobile robot.

Path planning algorithms must consider the actual size of the robot, so that the path planner is constrained to find an adequate collision-free trajectory. If the mobile manipulator is expected to move around, the arm should not be extended, so as to avoid recomputing the path while considering a change in the robot's footprint. When it does not carry any object, the arm resting position should be maintained. In the presence of an object, the manipulator should always be kept retracted so as to avoid unintentional contact/collision with other

objects or humans; the object should be kept in a proper position so as to reduce any risk of contact not only with humans, but also with some part of the robot itself.



Figure 5. Human (area modeled with a Gaussian-like distribution) walking near the robot.

In this way, the footprint of the mobile manipulator while it is in motion is always compliant with the original path planning, and there is no need to adapt the path based on the shape of the robot, thus reducing the computational effort each time the robot carries an object with a different shape or size.

3.3. Robot-to-Human Handover

In a context where humans and robots are actively collaborating to perform common operations, most of the interactions for the object handover are carried out from the human to the robot [54]. In handover processes, there are implicit and explicit communication interactions that should be interpreted, and this task is easier when it is a human who delivers something to a robotic system. However, in a complete HRC context, the object handover should be enabled in both directions, i.e., either from the human to the robot or vice versa. The latter one is more challenging, since the robot usually has limited communication mechanisms to make the object exchange; instead, humans generally have many feedback mechanisms during interactions, eye contact, touch contact, and even verbal communication, to ensure that the object is correctly delivered. With those feedback signals, it is possible to adapt the delivery speed, the gesture to handle dangerous objects, and even handover objects in different dynamic situations, e.g., walking by, sitting or standing [20]. Note that most gestures can be identified using skeletal data, point clouds, and wearable sensors; however, most of the approaches prefer the use of RGB videos due to their versatility in training recognition models [55].

In [56], a framework that enables a robot-to-human handover is presented. YOLO and GraspNet are used to identify the object and the grasping pose, respectively, while MediaPipe is used to detect the human hand to deliver the object. In such a framework, the mobile manipulator manages to deliver the objects most of the time; however, the dataset used for the tests is limited to a few objects, and it requires a lot of computational resources.

A UFactory xArm6 cobot is used in [57] to test the robot-to-human handover with the user wearing a vibrotactile device. In particular, such a device communicates with the user about the robot's intention to deliver an object. The aim was to improve human reaction in receiving the object while reducing his/her attention towards the robot.

In [38], a guided contact handover approach is proposed. The robot-to-human handover system tries to maximize the visibility and reachability of the object's contact part, which is defined by human comfort preferences. To this end, initial grasping positions are generated by Contact-GraspNet, and then the robot adjusts the pose to minimize the distance between the delivering object and the human.

In our previous work in [39], the robot handover is enabled by using gestures: the robot moves and releases the object once a specific gesture is recognized. The human hand gesture and its pose estimation are handled by MediaPipe, while the handover is executed by delivering the object towards a point near the human hand, as shown in Figure 6. In particular, the robot-to-human handover considers safety measures for the human by orienting the object's handle upwards and the dangerous part towards the ground.

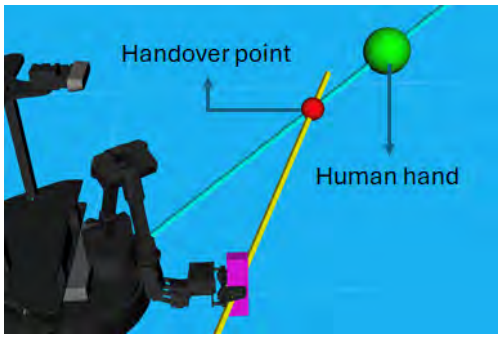


Figure 6. The handover path and the handover point.

In our framework, once the mobile manipulator successfully grasps the desired object and arrives at the delivery destination (where the user is located), the object should be handed over to the human as safely as possible. During that interaction, the handover planning could be constrained to be as human-like as possible, with the aim of making the user understand the intentions of the robot and retrieve the required item safely.

3.4. Intuitive User Interface

Most of the robots come with an interface to interact with them. For example, for industrial manipulators, the so-called *teach pendants* are generally used to monitor and control the motion of the robotic arm. This kind of device is widely used in industry, since it is designed for safe engagement with the user. However, the use of teach pendants may be not intuitive for non-skilled users or inexperienced human operators [58]. In [59], a human interface device is designed as an alternative to control a manipulator, allowing the user to intuitively guide the robot with the hand's kinesthetic sensations. Other interfaces use cameras that track human hand gestures and guide the robot, depending on the hand sign sequences [60].

In the ROS framework, it is possible to establish some interaction using graphical interfaces like Gazebo [61] and RViz [62], but those alone still need inputs from a terminal; in addition, the user should be trained to correctly command the robot through such interfaces. In general, it is important to choose the type of input and interaction that makes the application most feasible and comfortable for the user [63,64].

The functionalities of our proposed framework perform low-level tasks, such as navigation with obstacle avoidance, human identification and tracking, object identification, object grasping pose estimation, handover planning, etc. It is then important to combine all of them into an application, allowing the user to have full access to the robot's capabilities from a high-level point of view, as previously envisioned in [8].

On the basis of the preliminary results in [65], a user interface called BotBridge is here proposed; it is developed with Android Studio [66] to collect the input required to command the robot in a user-friendly way. The communication between the application and the robot is established using the Message Queuing Telemetry Transport (MQTT) protocol [67,68] over TCP/IP channels. The application receives the input command from the user to enable the robot; in particular, it enables the process to look for requested items in the environment. A preview of the application interface is presented in Figure 7, where the layout was designed to be executed on a tablet.



Figure 7. An overview of the Android application interface. Top: Initial interface for the user. Bottom: Screen for the user to call the robot.

The application allows the user to select a list of tools or objects, such as a knife, a fork, a spoon, a pair of scissors, a remote control, and a mouse, as shown in Figure 8. The robot, after receiving the request, explores the area where the objects are located. Then, it proceeds to pick and deliver the requested item to the user. After successfully delivering the item, the user is allowed to continue to request additional items or conclude the tasks for the robot. In the case that the services of the robot are no longer needed, the robot will automatically return to its home position or recharging location.

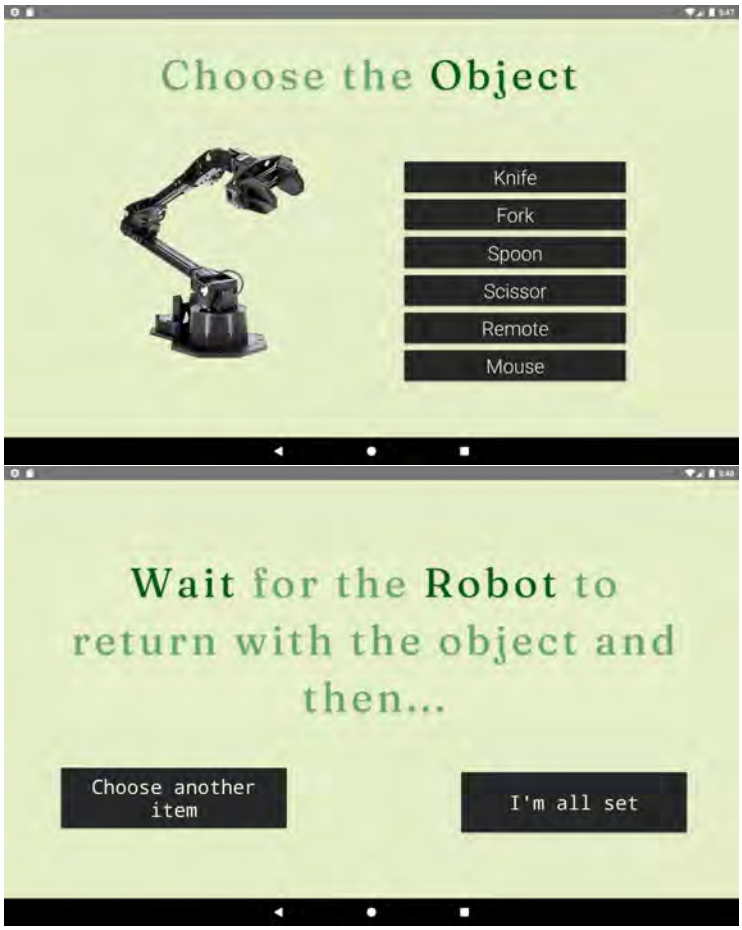


Figure 8. An overview of the Android application interface. **Top**: List of available items. **Bottom**: Screen to make further requests or conclude the service.

3.5. Framework Summary

After analyzing and comparing our previous works with existing solutions, the choices for our framework's functionalities are summarized in Table 2. Such choices consider the main features as well as the limitations of the algorithms to be implemented in a low-resource robotic system. Note that those algorithms are also expected to work in the near future, but they can be replaced by more recent approaches, thanks to the modularity of the framework. Further details about how they are connected will be explained in Section 4 along with the experimental setup.

Table 2. Summarizing table for our framework.

Functionality	Approach
Human detection	YOLOv8 [32]
Object instance segmentation for grasping pose estimation	YOLOv8s-seg [43]
Safe Navigation with humans	A* as a global planner with ROS social navigation layer [53]
Gesture recognition for handover	MediaPipe [69]
Intuitive user interface	Application with Android Studio [66,70]

4. Experimental Setup

In order to test and validate the proposed framework, a Locobot WX250 [71] mobile manipulator is used in a laboratory setup, where the space is shared with humans. The mobile manipulator, equipped with an RPLidar and an Intel RealSense D435 RGB-D camera,

is programmed within ROS1 Noetic in Ubuntu 20.04. Overall, the most recent framework integrates all the key functionalities designed in previous works, which are: (i) acceptance and processing of the user request, (ii) the robot's autonomous localization within the environment and search for the requested item, (iii) the robot's autonomous navigation towards the item location while considering safety distances from humans, (iv) the grasping procedure while considering the object's affordance, (v) maintenance of a safe pose while carrying the item, (vi) safe delivery of the requested item, which can be controlled using hand gestures in the case of robot-to-human handover. All such functionalities have been packed in the BotBridge application; the workflow to test the framework is illustrated in Figure 9, which is based on a work presented in [65]. A video showcasing the framework's execution is available in [72], while the GitHub repository is available in [73].

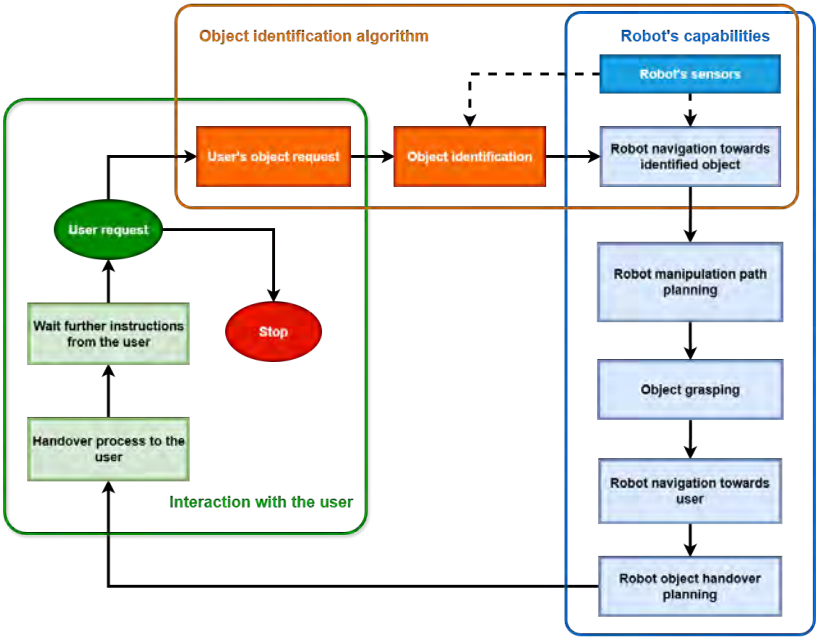


Figure 9. An overview of the framework workflow. Dashed lines indicate the data flow from the robot's sensors.

The mobile manipulator implements Real-Time Appearance-Based Mapping (RTAB-Map) [74] as the SLAM algorithm, while the ROS Navigation Stack [75] contains the A* algorithm as the global planner. For object detection, although there are more recent versions of YOLO, YOLOv8 has been chosen due to its fast image processing speed that favours real-time applications [76]. In particular, YOLOv8 has been used to detect people and estimate their position along with the ROS social navigation layer [53] to ensure safety when the robot moves near a human. Such a layer modifies the local plan to avoid humans safely through a proxemic layer, which associates the area occupied by humans with a Gaussian cost, proportional to their velocities. For the object detection and grasping pose estimation component, the three alternative approaches discussed in Section 3.1 were evaluated and compared: the bounding box-based detection with human-in-the-loop correction [33,34], the instance segmentation-based method using YOLOv8s-seg [43,73], and the affordance keypoint estimation approach using YOLOv8n-Pose [39,40]. Each implementation was tested within the complete framework, and the results are discussed hereafter.

Once the robot succeeds in grasping the object, before moving around the environment to deliver the requested item, the manipulator is set to a "safe position" to avoid any unintentional collision with humans and other objects. The described position is shown in Figure 10. As discussed in the previous section, the manipulator's pose does not increase

the robot's footprint, and the dangerous part points towards the ground. In this way, also during the phase of handover to the human, the robot is able to extend the item with the handle pointing towards the user.



Figure 10. Safe position of the manipulator after picking up the requested item. Lateral view (**left**) and top view (**right**).

The gesture recognition system is trained using MediaPipe [69]. In particular, the thumbs-up gesture is used to indicate to the robot that the user has picked the object, and the robot can release such an object. After successfully releasing the item, the robot returns to its safe position and awaits further instructions from the user.

The application starts when the user requests a mobile manipulator to retrieve an object from a given list and deliver it. Upon selecting the desired item, the mobile manipulator starts looking for the requested object within the environment. The starting point of the workflow in a real environment is depicted in Figure 11, while Figure 12 shows the object selection made by the user from the list. In this example, the user requested a scissor.



Figure 11. An illustration of the starting conditions for the framework.

The location of the objects can be a priori-defined; however, their exact position must be found by the robot once it reaches the objects' location. In our case, the objects are placed in a storage cabinet in different positions: laid down, hung, etc. Figure 13 shows the object identification process of the available tools when the robot proceeds to pick the one requested by the user.

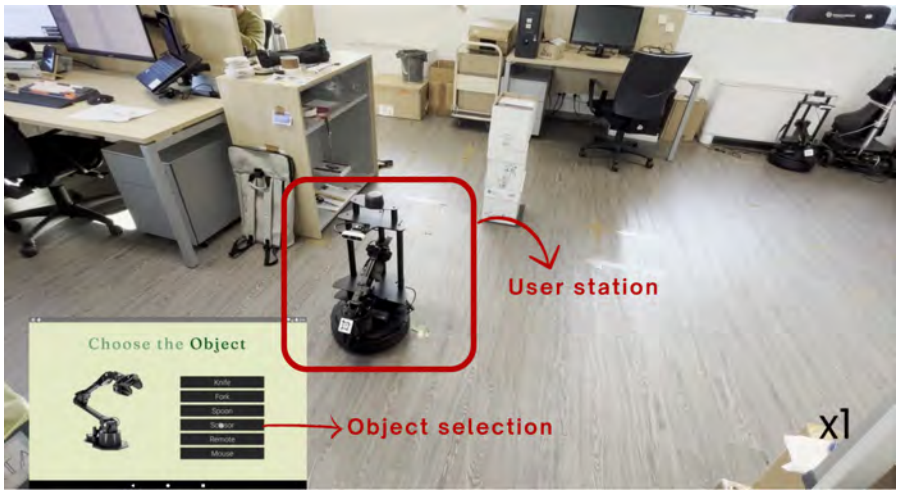


Figure 12. A view of item selection from the user.

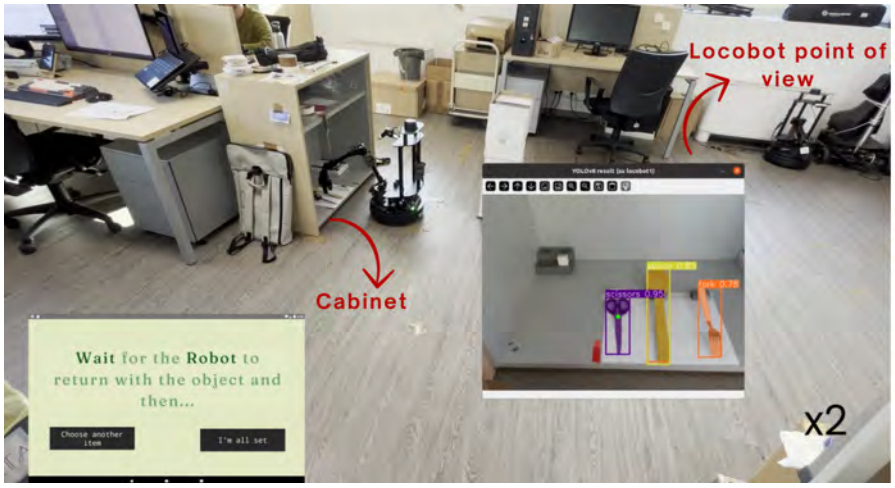


Figure 13. Item searching and identification in the cabinet.

The mobile manipulator implements a social navigation layer to safely avoid any human who is possibly encountered while the robot is moving (Figure 14).

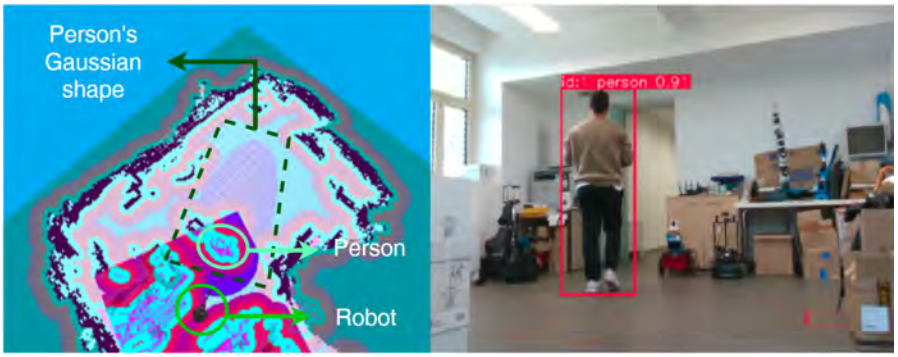


Figure 14. Safe navigation behavior of the mobile manipulator. (**Left**) RViz visualization of the person's Gaussian shape. (**Right**) Robot's point of view of the identified person.

Once the robot reaches the location of the storage cabinet, it uses the camera tilt to scan the space within the cabinet and look for the requested object. The grasping pose of the object is computed considering the algorithms detailed in Section 3, which consider the position of the center of mass of the segmented object and its orientation. Considering the object grasping pose as the goal for the manipulator's end effector, the path is computed to reach the object, grasp it, and prepare its delivery to the user.

As previously discussed, the arm grasping the object is put in the predefined "safety pose": the manipulator retracts itself within its footprint while carrying the object, orienting the tool (or the dangerous part of the object) towards the robot itself. In this way, the robot moves within the environment, respecting the navigation constraints.

As can be seen in Figure 15, once the robot reaches the user's location to deliver the object, the arm is extended with the object's handle pointing towards the user, while the dangerous part is always kept oriented towards the robot. The delivery is considered successful when the user picks the object, and he/she uses the thumbs-up gesture to communicate to the robot that the object has been delivered. After recognizing the gesture, the gripper releases the object, and the robot awaits for further instructions, as shown in Figure 16. If the services of the robot are no longer needed, the user concludes the robot's tasks; the mobile manipulator then returns to its home position or charging station, waiting to be called again. A more detailed workflow of the application within the proposed framework is illustrated in Figure 17.

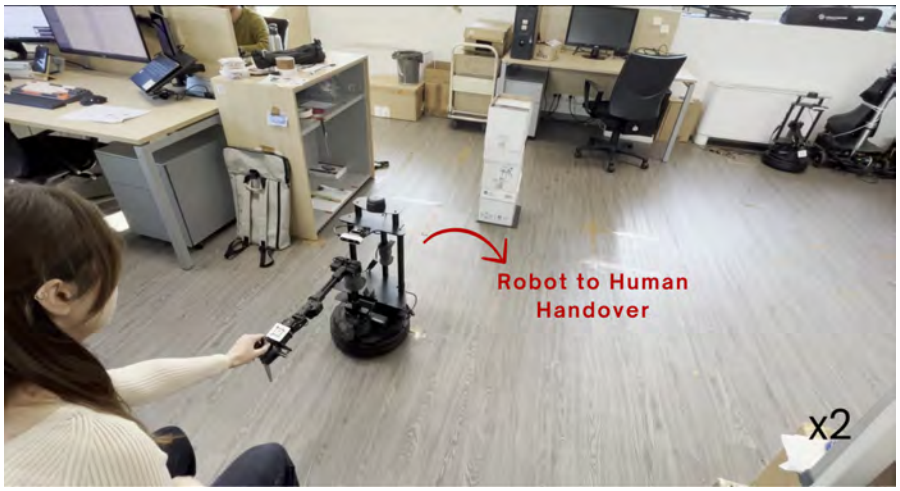


Figure 15. Robot-to-human item handover process.



Figure 16. The user shows the thumbs-up gesture to communicate the success of the delivery to the robot.

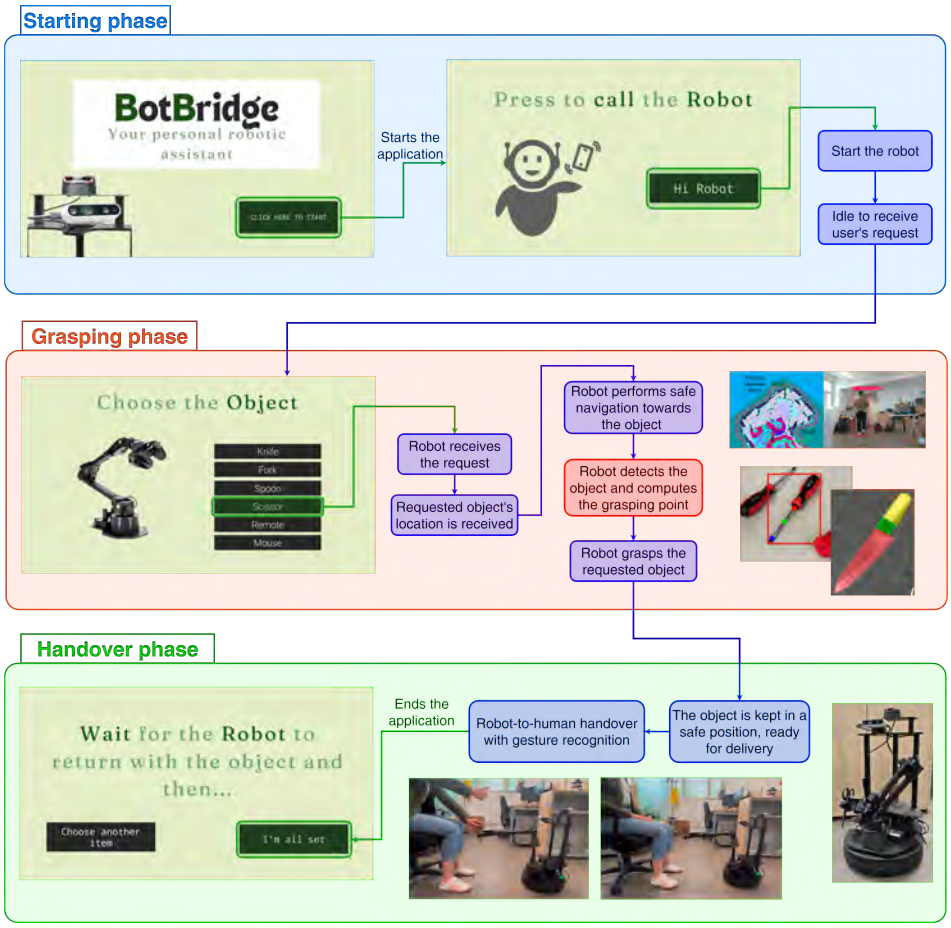


Figure 17. An overview of the application workflow.

Remark 1. *The framework is modular and specific algorithms for some functional blocks could be chosen according to the context. For example, in a structured environment, some objects and tools may be already positioned in a specific way, e.g., with their dangerous parts having a predefined, known orientation. In that case, it would be possible to simply use object segmentation to identify the object's grasping point with less focus on the object's grasping orientation. On the other hand, if the workspace is cluttered, it would be better to employ a more complex approach, such as the one presented in [33,39], since it provides the framework with additional flexibility to handle different objects with different poses.*

5. Experimental Validation

The proposed framework was evaluated in a real-world laboratory environment, where a human operator interacts with the Locobot WX250 mobile manipulator to request, receive, and use different objects. The complete workflow was successfully carried out in 15 trials: six of them were deployed in a static environment to focus on object retrieval and handover performance, while the remaining nine took place in a dynamic environment in which the robot shared space with humans. A video showcasing the complete execution is available in [72], and the source code is publicly accessible in [73]. For completeness, the computation times reported hereafter refer to the time needed to run each algorithm on the robot computer. This choice was made to highlight the applicability of the proposed framework to low-resource robots.

As described in Section 3, the framework requires a module for estimating a grasp pose that minimizes safety risks during object delivery. Implementations of the alternatives presented in Section 3.1 were evaluated, analyzing their main advantages and drawbacks.

The first implementation relied on instance segmentation [42], where the grasp point was computed as the 3D projection of the segmented mask's center of mass. Using the YOLOv8s-seg model with pre-trained Ultralytics weights [43] resulted in an inference time of approximately 480 ms, corresponding to about 2 FPS. While detection and segmentation performance are consistent with those reported in the literature, the relatively high inference time limits responsiveness. More importantly, the method does not explicitly encode information about dangerous object parts. As a result, although objects could generally be grasped successfully, the orientation of hazardous regions could not be reliably controlled during transport and handover, occasionally leaving the dangerous portion of the object exposed to the operator.

To improve responsiveness and grasp reliability in less structured environments, the bounding-box-based approach combined with human-in-the-loop correction [33] was evaluated. The YOLOv5s detector trained on a custom dataset of laboratory items achieved a mAP@50–90 of 0.95, although some overfitting was observed. By relying only on bounding box predictions, inference time was reduced to approximately 140 ms (∼7 FPS). While bounding boxes alone provide limited information about object geometry, the additional human feedback allowed the operator to correct the grasp position, ensuring stable grasps and safe manipulation even in unstructured scenes. This approach improved robustness but introduced continuous user intervention, partially conflicting with the objective of minimizing operator workload.

The third alternative employed an affordance keypoint estimation strategy [39]. A YOLOv8n-Pose model was trained to predict three affordance keypoints corresponding to the handle, the optimal robot grasping region, and the dangerous part of each object. This approach enables fully autonomous grasp planning while explicitly encoding safety-relevant object features. The model achieved a mAP@50–90 of 0.92 for bounding boxes and 0.88 for keypoints. Although these values likely reflect some degree of overfitting due to the limited dataset size, they indicate strong detection performance. The inference time was approximately 160 ms (∼6 FPS), enabling fully autonomous grasping and safe manipulation in moderately unstructured environments. Compared to the human-in-the-loop solution, this method reduces operator involvement at the cost of reduced customization of the handover configuration.

The safe navigation component was also evaluated. We conducted nine trials in which a human passed near the robot: three from the front, three from the side, and three from behind. The social navigation layer was implemented using a Gaussian-shaped costmap whose spread scales with human velocity [52]. Detection of humans, as well as their positions and velocities, was obtained by fusing data coming from the RGB-D and LiDAR sensors. This ensured reliable tracking during all trials. A minimum distance radius of 80 cm was enforced for stationary humans, corresponding to an effective minimum robot–human clearance of 60 cm when accounting for the robot's 20 cm radius. For moving human operators, the shape of their occupancy region was deformed along their motion direction using a Gaussian shape. The maximum deformation observed for a person walking fast led to an effective robot–human distance of 1.6 m. It is worth noting that the minimum robot–human clearance of 60 cm does not represent a hard constraint. Indeed, the planner can create a path for the robot that falls inside the human's occupancy shape, but the cost increases as the robot gets closer to the human. In this way, the planner was able to balance path optimality and human distance. This resulted in a measured minimum human–robot distance of 30 cm in worst-case trial scenarios. To evaluate the reactivity of the local replanning, selected trials required a human to walk alongside the robot that was already moving. In these scenarios, replanning was triggered multiple times in real time, and the planner exhibited smooth trajectory adaptation that would have guided the robot outside the human's occupancy area.

The handover stage relies on gesture-based human initiation using MediaPipe. Relying only on the Locobot's computer, the system operates with an inference time of approximately 50 ms (∼20 FPS), enabling real-time hand detection. During the experiments, the robot successfully approached the detected hand position, while maintaining a predefined safety distance of 20 cm, stopping before entering the operator's immediate space. The final grasp of the handle was therefore performed by the human, thus reducing the likelihood of unexpected human–robot contact. Thanks to the ∼95% detection accuracy of MediaPipe, as reported in the literature, the probability of unintended handover initiation or premature object release remains low.

For what concerns the usability of the BotBridge application, we collected feedback from seven users to assess the functionality of its user interface. All participants had a moderately technical background, which may introduce bias in the findings. Overall, users evaluated the interface positively, emphasizing its simplicity and intuitive design. In particular, they appreciated the minimalistic layout, which supported efficient task completion without unnecessary visual or interactional complexity. The mobile-oriented design (smartphones and tablets) also contributed to a sense of familiarity, as it leveraged interaction patterns common in everyday applications. Some participants, however, reported that interacting with the system via a touchscreen could disrupt their workflow and reduce their focus on the primary task. To mitigate this limitation, several users proposed the integration of voice commands to enable more seamless, hands-free interaction, which can be considered as a future improvement.

Remark 2. *A direct quantitative comparison with other frameworks is not straightforward, since, to the best of the authors' knowledge, no existing system simultaneously addresses all the considered functionalities within a single mobile manipulation platform. Comparisons can only be drawn at the component level, as in Table 1. Some approaches in the literature achieve better performance on specific tasks [27,35,36,51] but rely on more powerful hardware, including dedicated GPUs or multi-camera setups. The hardware constraints of the adopted platform have guided the algorithm selection summarized in Table 2, prioritizing deployability and modularity over high performance.*

6. Conclusions and Future Works

This paper reviewed all the main functionalities required to develop a low-resource but effective robotic assistant system, proposing as a result a complete framework enabling safe interaction between humans and robots, with particular focus on safe robot-to-human object handover. The current framework is an evolution of previous works, where specific features were developed separately and then combined to obtain a complete workflow.

In particular, the framework was developed targeting low-resource robots that do not have high-end sensors, computers, processors, dedicated GPUs, etc. However, the proposed approach can be adapted and generalized to higher-performing systems, but the purpose of the authors was to provide a complete baseline for safe HRI, potentially implementable in small and medium enterprises, which suffer most from technological migration due to their limited resources. Moreover, even though the algorithms were deployed in a mobile manipulator, it is possible to reuse the basic operations proposed in the framework in decoupled systems, e.g., grasping and handover in manipulators and safe navigation in mobile robots.

It must be noted that few users have tested the BotBridge application so far; the significance of their feedback could then be limited and insufficient to provide a complete evaluation of the application's performance in terms of usability and intuitiveness. In this regard, increasing the number of users could allow the collection of suitable data for enhancing the application and improving the overall framework's performance.

Future works might include an additional system that monitors the human and robot's working space to cover larger areas, as well as in other application contexts, while ensuring safety and providing redundancy in the case that the sensors of the robot are occluded or limited. Furthermore, management of unexpected dynamic obstacles during the robot-to-human handover phase could be investigated to further guarantee safety and applicability in human-shared environments.

Moreover, despite most of the industrial setups preferring non-verbal communication with robots (due to noise, distracting factors, etc.), in other contexts having additional feedback from the robot might improve the user's experience while collaborating with the robots.

Author Contributions: Conceptualization, P.D.C.C.; methodology, P.D.C.C., C.L.B., R.F.C. and A.R.; software, C.L.B., R.F.C. and A.R.; validation, C.L.B., R.F.C. and A.R.; formal analysis, P.D.C.C. and M.I.; investigation, P.D.C.C., C.L.B., R.F.C., A.R. and M.I.; resources, M.I.; data curation, P.D.C.C.; writing—original draft preparation, P.D.C.C.; writing—review and editing, P.D.C.C., C.L.B., R.F.C., A.R. and M.I.; visualization, P.D.C.C., C.L.B., R.F.C. and A.R.; supervision, P.D.C.C. and M.I.; project administration, P.D.C.C. and M.I.; funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. Written informed consent has been obtained from the subjects to publish this paper.

Data Availability Statement: The code used to implement the proposed framework (partial and whole) is available in GitHub repositories: [31,34,40,73].

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
HRI	Human–Robot Interaction
ISO	International Organization for Standardization
ISO/TS	International Organization for Standardization/Technical Specification
cobot	Collaborative robot
RGB-D	Red, Green, Blue and Depth
ROS	Robot Operating System
DWA	Dynamic Window Approach
TEB	Timed Elastic Band
GTEB	Goal-oriented Timed Elastic Band
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ARTag	Augmented Reality Tag
ArUco	Augmented Reality University of Cordoba
YOLO	You Only Look Once
PRM	Probabilistic Roadmap
RRT	Rapidly exploring random tree
LIDAR	Light Detection and Ranging
HRC	Human–Robot Collaboration
RViz	Robot Visualizer
RTAB-Map	Real-Time Appearance-Based Mapping

References

1. Islam, M.T.; Sepanloo, K.; Woo, S.; Woo, S.H.; Son, Y.J. A review of the industry 4.0 to 5.0 transition: Exploring the intersection, challenges, and opportunities of technology and human–machine collaboration. *Machines* **2025**, *13*, 267. [CrossRef]

2. Firmino de Souza, D.; Sousa, S.; Kristjuhan-Ling, K.; Dunajeva, O.; Roosileht, M.; Pentel, A.; Mõttus, M.; Can Özdemir, M.; Gratšjova, Ž. Trust and trustworthiness from human-centered perspective in human–robot interaction (HRI)—A systematic literature review. *Electronics* **2025**, *14*, 1557. [CrossRef]
3. Ding, P.; Zhang, J.; Zheng, P.; Zhang, P.; Fei, B.; Xu, Z. Dynamic scenario-enhanced diverse human motion prediction network for proactive human–robot collaboration in customized assembly tasks. *J. Intell. Manuf.* **2025**, *36*, 4593–4612. [CrossRef]
4. Sun, Y.; Jeelani, I.; Gheisari, M. Safe human-robot collaboration in construction: A conceptual perspective. *J. Saf. Res.* **2023**, *86*, 39–51. [CrossRef]
5. SMBPB, S.; Valori, M.; Legnani, G.; Fassi, I. Assessing safety in physical human–robot interaction in industrial settings: A systematic review of contact modelling and impact measuring methods. *Robotics* **2025**, *14*, 27. [CrossRef]
6. Memon, M.L.; Khan, M.N.; Shaikh, A.A. Industry 5.0: Human-Robot Interaction, Smart Manufacturing, and AI/ML Integration—A Comprehensive Review for Next-Generation Manufacturing Systems. *Spectr. Eng. Sci.* **2025**, *3*, 1889–1928.
7. Salami, M.; Bilancia, P.; Peruzzini, M.; Pellicciari, M. A framework for integrated design of human–robot collaborative assembly workstations. *Robot. Comput.-Integr. Manuf.* **2026**, *97*, 103108. [CrossRef]
8. Cen Cheng, P.D.; Sibona, F.; Indri, M. A framework for safe and intuitive human-robot interaction for assistant robotics. In *Proceedings of the 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.
9. Li, W.; Hu, Y.; Zhou, Y.; Pham, D.T. Safe human–robot collaboration for industrial settings: A survey. *J. Intell. Manuf.* **2024**, *35*, 2235–2261. [CrossRef]
10. Robla-Gómez, S.; Becerra, V.M.; Llata, J.R.; Gonzalez-Sarabia, E.; Torre-Ferrero, C.; Perez-Oria, J. Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access* **2017**, *5*, 26754–26773. [CrossRef]
11. Hamad, M.; Kurdas, A.; Mansfeld, N.; Abdolshah, S.; Haddadin, S. Modularize-and-conquer: A generalized impact dynamics and safe precollision control framework for floating-base tree-like robots. *IEEE Trans. Robot.* **2023**, *39*, 3200–3221. [CrossRef]
12. Abdulazeem, N.; Hu, Y. Human factors considerations for quantifiable human states in physical human–robot interaction: A literature review. *Sensors* **2023**, *23*, 7381. [CrossRef]
13. Panagou, S.; Neumann, W.P.; Fruggiero, F. A scoping review of human robot interaction research towards Industry 5.0 human-centric workplaces. *Int. J. Prod. Res.* **2024**, *62*, 974–990. [CrossRef]
14. Marin Vargas, A.; Cominelli, L.; Dell'Orletta, F.; Scilingo, E.P. Verbal communication in robotics: A study on salient terms, research fields and trends in the last decades based on a computational linguistic analysis. *Front. Comput. Sci.* **2021**, *2*, 591164. [CrossRef]
15. Urakami, J.; Seaborn, K. Nonverbal cues in human–robot interaction: A communication studies perspective. *ACM Trans. Hum.-Robot Interact.* **2023**, *12*, 1–21. [CrossRef]
16. Tiferes, J.; Hussein, A.A.; Bisantz, A.; Higginbotham, D.J.; Sharif, M.; Kozlowski, J.; Ahmad, B.; O'Hara, R.; Wawrzyniak, N.; Guru, K. Are gestures worth a thousand words? Verbal and nonverbal communication during robot-assisted surgery. *Appl. Ergon.* **2019**, *78*, 251–262. [CrossRef]
17. Saunderson, S.; Nejat, G. How robots influence humans: A survey of nonverbal communication in social human–robot interaction. *Int. J. Soc. Robot.* **2019**, *11*, 575–608. [CrossRef]
18. Secil, S.; Ozkan, M. A collision-free path planning method for industrial robot manipulators considering safe human–robot interaction. *Intell. Serv. Robot.* **2023**, *16*, 323–359. [CrossRef]
19. Liu, H.; Wang, L. Collision-free human-robot collaboration based on context awareness. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 101997. [CrossRef]
20. Kupcsik, A.; Hsu, D.; Lee, W.S. Learning dynamic robot-to-human object handover from human feedback. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 1, pp. 161–176.
21. Costanzo, M.; De Maria, G.; Natale, C. Handover control for human-robot and robot-robot collaboration. *Front. Robot. AI* **2021**, *8*, 672995. [CrossRef] [PubMed]
22. Al, G.A.; Martinez-Hernandez, U. Safe multi-channel communication for human–robot collaboration. *Robot. Comput.-Integr. Manuf.* **2026**, *97*, 103109. [CrossRef]
23. Faibish, T.; Kshirsagar, A.; Hoffman, G.; Edan, Y. Human preferences for robot eye gaze in human-to-robot handovers. *Int. J. Soc. Robot.* **2022**, *14*, 995–1012. [CrossRef]
24. Ou, X.; You, Z.; He, X. Local Path Planner for Mobile Robot Considering Future Positions of Obstacles. *Processes* **2024**, *12*, 984. [CrossRef]
25. Ngo, T.D.; Truong, X.T. Socially aware robot navigation framework: Where and how to approach people in dynamic social environments. *IEEE Trans. Autom. Sci. Eng.* **2022**, *20*, 1322–1336. [CrossRef]
26. Yamabata, Y.; Venture, G. User Perception of Socially-Aware Robot Navigation with Engagement-Based Proxemics. In *Proceedings of the 2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1408–1414.

27. Qichao, J.; Ali, H.B.B.Y.; Chamran, M.K. Learning Socially Compliant Navigation with a Proxemics-Informed Composite Reward Function. In *Proceedings of the 2025 IEEE 9th International Conference on Software Engineering & Computer Systems (ICSECS)*; IEEE: Piscataway, NJ, USA, 2025; pp. 606–610.
28. Chen, Y.; Yang, C.; Gu, Y.; Hu, B. Influence of mobile robots on human safety perception and system productivity in wholesale and retail trade environments: A pilot study. *IEEE Trans. Hum.-Mach. Syst.* **2022**, *52*, 624–635. [CrossRef]
29. Kalaitzakis, M.; Cain, B.; Carroll, S.; Ambrosi, A.; Whitehead, C.; Vitzilaios, N. Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers. *J. Intell. Robot. Syst.* **2021**, *101*, 71. [CrossRef]
30. Cen Cheng, P.D.; Indri, M.; Maresca, F.; Ragazzo, A.; Sibona, F. A software architecture for low-resource autonomous mobile manipulation. In *Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8.
31. Ragazzo, A.; Maresca, F. GitHub Repository for A Software Architecture for Low-Resource Autonomous Mobile Manipulation. Available online: https://github.com/AntoRag/thesis (accessed on 1 February 2026).
32. Ultralytics. YOLOv8 Models—Ultralytics Documentation. 2025. Available online: https://docs.ultralytics.com/models/yolov8/ (accessed on 1 September 2025).
33. Cavelli, R.F.; Cen Cheng, P.D.; Indri, M. Motion Planning and Safe Object Handling for a Low-Resource Mobile Manipulator as Human Assistant. In *Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.
34. GitHub Repository for Motion Planning and Safe Object Handling for a Low-Resource Mobile Manipulator as Human Assistant. Available online : https://github.com/Saro0800/296846-MasterThesis.git (accessed on 1 April 2024).
35. Fang, H.S.; Wang, C.; Gou, M.; Lu, C. Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11444–11453.
36. Fang, H.S.; Wang, C.; Fang, H.; Gou, M.; Liu, J.; Yan, H.; Liu, W.; Xie, Y.; Lu, C. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Trans. Robot.* **2023**, *39*, 3929–3945. [CrossRef]
37. Murali, A.; Sundaralingam, B.; Chao, Y.W.; Yamada, J.; Yuan, W.; Carlson, M.; Ramos, F.; Birchfield, S.; Fox, D.; Eppner, C. GraspGen: A Diffusion-based Framework for 6-DOF Grasping with On-Generator Training. *arXiv* **2025**, arXiv:2507.13097.
38. Wang, Z.; Liu, Z.; Ouporov, N.; Song, S. ContactHandover: Contact-Guided Robot-to-Human Object Handover. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 9916–9923.
39. Blengini, C.L.; Cen Cheng, P.D.; Indri, M. Safe robot affordance-based grasping and handover for Human-Robot assistive applications. In *Proceedings of the IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
40. Github Repository for Safe Robot Affordance-Based Grasping and Handover for Human-Robot Assistive Application. Available online: https://github.com/celubi/affordance_based_handover_grasping.git (accessed on 1 May 2024).
41. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. [CrossRef]
42. Ultralytics. Semantic Segmentation—Ultralytics Documentation. 2025. Available online: https://www.ultralytics.com/glossary/semantic-segmentation (accessed on 1 September 2025).
43. Ultralytics. Segmentation Task—Ultralytics Documentation. 2025. Available online: https://docs.ultralytics.com/it/tasks/segment/ (accessed on 1 September 2025).
44. Kobayashi, M.; Motoi, N. Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles. *IEEE Access* **2022**, *10*, 17018–17029. [CrossRef]
45. Rösmann, C.; Hoffmann, F.; Bertram, T. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In *Proceedings of the 2015 European Control Conference (ECC)*; IEEE: Piscataway, NJ, USA, 2015; pp. 3352–3357.
46. Dang, T.V. Autonomous mobile robot path planning based on enhanced A* algorithm integrating with time elastic band. *MM Sci. J.* **2023**, *2023*, 6717–6722. [CrossRef]
47. Liu, Y.; Wang, C.; Wu, H.; Wei, Y. Mobile robot path planning based on kinematically constrained A-star algorithm and DWA fusion algorithm. *Mathematics* **2023**, *11*, 4552. [CrossRef]
48. Pimentel, F.d.A.M.; Aquino-Jr, P.T. Evaluation of ROS navigation stack for social navigation in simulated environments. *J. Intell. Robot. Syst.* **2021**, *102*, 87. [CrossRef]
49. Singamaneni, P.T.; Bachiller-Burgos, P.; Manso, L.J.; Garrell, A.; Sanfeliu, A.; Spalanzani, A.; Alami, R. A survey on socially aware robot navigation: Taxonomy and future challenges. *Int. J. Robot. Res.* **2024**, *43*, 1533–1572. [CrossRef]
50. Alyassi, R.; Cadena, C.; Riener, R.; Paez-Granados, D. Social robot navigation: A review and benchmarking of learning-based methods. *Front. Robot. AI* **2025**, *12*, 1658643. [CrossRef]
51. Linh, K.; Cox, J.; Buiyan, T.; Lambrecht, J. All-in-one: A drl-based control switch combining state-of-the-art navigation planners. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2022; pp. 2861–2867.

52. Cen Cheng, P.D.; Indri, M.; Maresca, F.; Ragazzo, A.; Sibona, F. Dynamic path planning in human-shared environments for low-resource mobile agents. In *Proceedings of the 2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
53. ROS. *ROS Social Navigation Layers*. Available online: https://wiki.ros.org/social_navigation_layers (accessed on 1 September 2025).
54. Duan, H.; Yang, Y.; Li, D.; Wang, P. Human–robot object handover: Recent progress and future direction. *Biomim. Intell. Robot.* **2024**, *4*, 100145. [CrossRef]
55. Guan, S.; Wang, J.; Wang, X.; Ding, C.; Liang, H.; Wei, Q. Dynamic gesture recognition during human–robot interaction in autonomous earthmoving machinery used for construction. *Adv. Eng. Inform.* **2025**, *65*, 103315. [CrossRef]
56. Meng, C.; Zhang, T.; Zhao, D.; Lam, T.L. Fast and Comfortable Robot-to-Human Handover for Mobile Cooperation Robot System. *Cyborg. Bionic. Syst.* **2024**, *5*, 0120. [CrossRef]
57. Mohammed Zaffir, M.A.B.; Wada, T. Presentation of robot-intended handover position using vibrotactile interface during robot-to-human handover task. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, Boulder, CO, USA, 11–15 March 2024; pp. 492–500.
58. Safeea, M.; Neto, P. Precise positioning of collaborative robotic manipulators using hand-guiding. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 5497–5508. [CrossRef]
59. Lee, D.; Hwang, J.; Jung, D.; Koh, J.S.; Do, H.; Kim, U. Intuitive six-degree-of-freedom human interface device for human-robot interaction. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 7507310. [CrossRef]
60. Abhishek, S.; Jogi, Y.S.; Sahu, U.K.; Dash, S.K.; Yadav, U.K. Teach Pendant at Fingertips: Intuitive Vision-based Gesture-Driven Control of Dexter ER2 Robotic Arm. *IEEE Access* **2025**, *13*, 100614–100629. [CrossRef]
61. Gazebo Classic. Guided Tutorial: Beginner Level 1. 2024. Available online: https://classic.gazebosim.org/tutorials?tut=guided_b1 (accessed on 8 January 2026).
62. Meegada, P. Getting Started with RViz: A Beginner's Guide to ROS Visualization. 2024. Available online: https://medium.com/@pranathi.meegada/getting-started-with-rviz-a-beginners-guide-to-ros-visualization-2e5f4156e410 (accessed on 8 January 2026).
63. Frijns, H.A.; Schmidbauer, C. Design guidelines for collaborative industrial robot user interfaces. In *Proceedings of the IFIP Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 407–427.
64. Dall'Alba, D.; Boriero, F. Towards an intuitive industrial teaching interface for collaborative robots: Gamepad teleoperation vs. kinesthetic teaching. *Int. J. Adv. Manuf. Technol.* **2025**, *138*, 1505–1522 [CrossRef]
65. Ripi, A. Development of a Human-Centered Framework for Safe Object Handling with Mobile Manipulators. Master's Thesis, Politecnico di Torino, Turin, Italy, 2024.
66. Android Developers. Introduzione ad Android Studio. 2024. Available online: https://developer.android.com/studio/intro?hl=it (accessed on 1 September 2024).
67. MQTT.org. MQTT—The Standard for IoT Messaging. 2025. Available online: https://mqtt.org/ (accessed on 1 September 2025).
68. OASIS. MQTT Version 3.1.1. 2025. Available online: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html (accessed on 1 September 2025).
69. Google AI. MediaPipe Vision: Gesture Recognizer. 2025. Available online: https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer?hl=it (accessed on 1 December 2025).
70. Android Developers. Your First Kotlin Program on Android. 2024. Available online: https://developer.android.com/kotlin/first (accessed on 1 September 2024).
71. Trossen Robotics. Interbotix XSLocoBot Specifications. 2026. Available online: https://docs.trossenrobotics.com/interbotix_xslocobots_docs/specifications.html# (accessed on 8 January 2026).
72. Video. Video Demo for Framework's Experimental Testing. 2024. Available online: https://www.youtube.com/watch?v=wsEJ8mzYE2Q (accessed on 8 January 2026).
73. Ripi, A. GitHub Repository for a Framework for Safe Mobile Manipulation in Human-Centered Applications. Available online: https://github.com/angelaripi/Locobot-Project/tree/main (accessed on 8 January 2026).
74. Labbé, M.; Michaud, F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. Field Robot.* **2019**, *36*, 416–446. [CrossRef]
75. Trossen Robotics. Interbotix XSeries Locobots—Navigation Stack Configuration. 2026. Available online: https://docs.trossenrobotics.com/interbotix_xslocobots_docs/ros1_packages/navigation_stack_configuration.html (accessed on 8 January 2026).
76. Sapkota, R.; Karkee, M. Comparing YOLOv11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. *arXiv* **2024**, arXiv:2410.19869.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.