applied sciences

MDPI

Article

Integration of Deep Learning and Collaborative Robot for Assembly Tasks

Enrico Mendez [†], Oscar Ochoa [†], David Olivera-Guzman, Victor Hugo Soto-Herrera, José Alfredo Luna-Sánchez, Carolina Lucas-Dophe, Eloina Lugo-del-Real, Ivo Neftali Ayala-Garcia, Miriam Alvarado Perez and Alejandro González [*]

Tecnologico de Monterrey, School of Engineering and Sciences, Santiago de Querétaro 76130, Querétaro, Mexico; enrico.mendez@outlook.es (E.M.); oscar.ochoa4242@gmail.com (O.O.); david.oliverag@gmail.com (D.O.-G.); victor.hugosoto229@gmail.com (V.H.S.-H.); alfredo.2001.luna@gmail.com (J.A.L.-S.); carolinalucasdophe011200@gmail.com (C.L.-D.); eloina.ldelreal@tec.mx (E.L.-d.-R.); ivo.ayala@tec.mx (I.N.A.-G.); miriam.alvarado@tec.mx (M.A.P.)

* Correspondence: a.gonzalezda@tec.mx

† These authors contributed equally to this work.

Abstract: Human–robot collaboration has gained attention in the field of manufacturing and assembly tasks, necessitating the development of adaptable and user-friendly forms of interaction. To address this demand, collaborative robots (cobots) have emerged as a viable solution. Deep Learning has played a pivotal role in enhancing robot capabilities and facilitating their perception and understanding of the environment. This study proposes the integration of cobots and Deep Learning to assist users in assembly tasks such as part handover and storage. The proposed system includes an object classification system to categorize and store assembly elements, a voice recognition system to classify user commands, and a hand-tracking system for close interaction. Tests were conducted for each isolated system and for the complete application as used by different individuals, yielding an average accuracy of 91.25%. The integration of Deep Learning into cobot applications has significant potential for transforming industries, including manufacturing, healthcare, and assistive technologies. This work serves as a proof of concept for the use of several neural networks and a cobot in a collaborative task, demonstrating communication between the systems and proposing an evaluation approach for individual and integrated systems.

Keywords: cobots; Deep Learning; neural network; collaborative robotics; human–robot interaction

check for updates

Citation: Mendez, E.; Ochoa, O.; Olivera-Guzman, D.; Soto-Herrera, V.H.; Luna-Sánchez, J.A.; Lucas-Dophe, C.; Lugo-del-Real, E.; Ayala-Garcia, I.N.; Alvarado Perez, M.; González, A. Integration of Deep Learning and Collaborative Robot for Assembly Tasks. *Appl. Sci.* **2024**, *14*, 839. https://doi.org/10.3390/app14020839

Academic Editor: Adel Razek

Received: 11 November 2023
Revised: 16 December 2023
Accepted: 17 December 2023
Published: 18 January 2024

1. Introduction

The manufacturing industry currently requires more flexible and user-friendly machinery, which has led to the growing popularity of collaborative robots (cobots) [1,2]. Cobots are specifically designed to work alongside humans, performing repetitive, dangerous, or high-precision tasks while also interacting and collaborating with humans. Their programming flexibility allows users to reconfigure and adapt them for various tasks, placing humans at the center of the manufacturing process and promoting harmonious collaboration between humans and machines [3].

Traditional industrial automation has undergone significant changes. Fixed automation systems and robotic arms have played a critical role in achieving high production rates and ensuring consistency in mass-production industries. However, these systems have limitations in their ability to adapt to evolving production needs and seamlessly collaborate with human workers. As manufacturing requirements continue to evolve, there is an increasing demand for more flexible and efficient solutions. Although humans can easily deal with the variability of the process, they are restricted by their physical capabilities in terms of speed, strength, precision, and repeatability [4]. Human limitations can hinder the efficiency of the manufacturing process; therefore, human–robot collaboration (HRC) has been implemented as a new approach in which humans and robots work together, taking advantage of automation and manual labor [5]. Thus, the limitations of traditional automation systems highlight the need for more versatile and adaptable technologies, such as cobots and machine learning [6–8]. In this context Cyber–physical systems have emerged as a new paradigm. These systems integrate software and physical hardware components and have the potential to automate tasks in almost every industry [9].

In recent years, various machine learning approaches have emerged to address problems of different magnitudes and complexities. For example, Deep Learning (DL) has gained significant attention and relevance in both research and industry due to its ability to make predictions based on extensive datasets used for training. DL algorithms are specifically designed to identify patterns and features within complex data, enabling machines to perform more intricate tasks than ever before [10]. DL algorithms have also been successfully applied to cobots to enhance their sensing capabilities, analyze visual information, design optimal trajectories, and improve control mechanisms, among other applications. The integration of DL in cobots has the potential to revolutionize industries across the board, including logistics, manufacturing, healthcare, and assistive technologies [11,12].

Previous studies have explored the implementation of DL in cobots for collaborative manufacturing and human–robot interactions [13]. These applications include palletization and assisted manufacturing [14], contact identification in pick-and-place routines [15], object handling with trust-aware human–robot interaction [16], collaborative assembly based on hand gestures [17], learning and reproducing human-taught activities [18], adaptive user preference-based robot assistance [19,20], and collaborative manufacturing tasks [21,22]. Moreover, significant efforts have been dedicated to harnessing artificial intelligence in the realm of medical robotics [23]. Noteworthy instances include the application of automated probe movement in obstetric ultrasound [24] and the development of placental pose estimation to facilitate automated fetoscopy [25].

Artificial Neural Networks have been extensively utilized in cobot applications for human–robot collaboration, serving various purposes. For example, voice-command recognition enables cobots to receive instructions through spoken commands related to movements, positions, or speed. This approach has been applied in domains such as medical assistance, workshops, and manufacturing [26–28].

Furthermore, computer vision systems integrated with neural networks have found multiple applications in cobots. One such application is image classification, specifically object recognition, which allows cobots to interact with different objects. This capability has been utilized for route control and pick-and-place routines [29,30]. Additionally, computer vision in cobots has been employed for safety purposes, including gesture recognition and human position tracking [31,32], where vision systems integrated with neural networks are applied to track the body of the user.

In a specific study by Liu et al. [33], a system consisting of multiple neural networks was developed for multi-modal human–robot collaboration, incorporating audio, gesture, and body position inputs. However, there was no direct interaction between the different DL systems utilized.

Through our examination of the current research on human–robot collaboration, we identified a notable absence of comprehensive systems able to process inputs from different sources that facilitate human-like interactions. Moreover, the limited instances of employing multiple Deep Learning models in human–robot applications were evident in our review [12,13].

This study seeks to address these gaps by developing a human–robot interaction system capable of processing inputs from a variety of sources. The primary aim of this study is to address the following research questions: (1) How can different Deep Learning models be integrated into a human–robot system for voice-command recognition, part classification, and hand-tracking in assisting assembly tasks? (2) What is the success rate of collaborative routines between humans and robots facilitated by Deep Learning integration for assembly activities involving part handover and storage? These questions guide our exploration into the feasibility and effectiveness of incorporating multiple Deep Learning models for enabling human-like collaborative activities in assembly tasks.

In this work, we propose the integration and application of cobots and DL to assist users in an assembly activity involving the handover and storage of parts. These two activities are supported by three Convolutional Neural Networks (CNNs). One CNN is focused on recognizing the 11 voice commands required to guide the robot in the desired activities; another CNN uses images to classify the parts received from the user. Finally, a pre-trained hand-tracking CNN facilitates the approach of the robot to the hand of the user for part handover and retrieval. Figure 1 illustrates the interconnected systems involved in each task. The rest of the paper is organized as follows. Section 2 provides information on the materials and methods used in each of the composing systems. Section 3 has a detailed description of the processes and methods applied for the stated problem. Section 4 describes the proposed experiments. Section 5 showcases the results, while Section 6 discusses and compares the results with the existing literature. Finally, Section 7 provides the conclusions drawn from the findings.

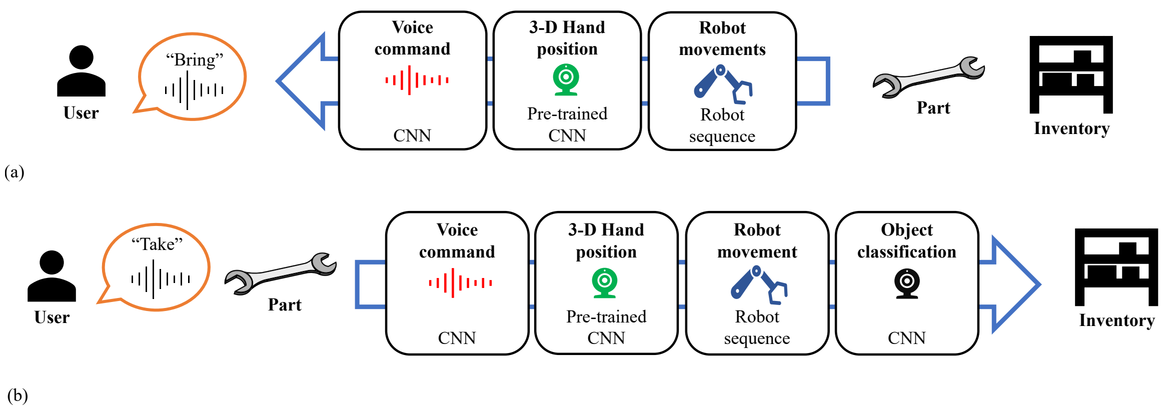


Figure 1. Processes involved in the functionality of the system. The arrow represents the part flow from or toward the user. (**a**) The user asks the robot to bring a part. (**b**) The user asks the robot to store a part.

2. Materials and Methods

2.1. Object Classification

In object classification, various strategies leverage computer vision techniques to categorize objects within images [34]. One common approach involves traditional computer vision methods, such as feature extraction and template matching [35]. These techniques rely on identifying distinctive patterns or features in images and comparing them to predefined templates for classification.

Another widely employed strategy is machine learning-based classification, where algorithms learn patterns and relationships from labeled data used for training [36]. Support Vector Machines (SVMs) and Random Forests are examples of such techniques. These models analyze input features and make predictions based on learned patterns [37].

Notably, CNNs represent a state-of-art strategy for object classification [38,39], and have steadily gained popularity in recent years. CNNs outperform other methods at automatically learning hierarchical features directly from pixel values, making them highly effective in direct image-related tasks. These neural networks employ convolutional layers to extract the spatial hierarchies of features, enabling robust object recognition and classification [40].

2.2. Voice Recognition

Voice recognition is a field that has evolved significantly over recent decades. Researchers have focused on various scientific techniques to enhance voice recognition systems. These techniques can be separated into two approaches, speaker-dependent and speaker-independent, each with distinct advantages and applications.

Speaker-dependent recognition involves training the system with samples from specific people and adapting the system to recognize them accurately. Its primary drawback lies in its limited applicability, catering only to certain users. If a user who has not trained the model attempts to use it, there is a high likelihood of commands being misclassified. On the contrary, speaker-independent recognition aims to identify speakers without prior training, making it a versatile and general solution [41].

Furthermore, neural networks have become a powerful tool for voice recognition. Deep Learning models, such as CNNs, have exhibited remarkable results in improving the accuracy of voice recognition systems. These networks excel at extracting meaningful features from audio data, thereby enhancing recognition performance and enabling the system to adapt and recognize diverse speakers more effectively [42].

CNNs' approach involves representing one-dimensional (1D) audio signals as 2D spectrograms. Before any analysis of audio samples, raw audio data are transformed into 2D spectrograms. This representation offers a visual representation of audio signals, facilitating more robust feature extraction and classification [43].

2.3. Human Body Tracking

Human body tracking involves monitoring and analyzing human movements within a given space. This can be conducted for body parts or the full body. Tracking body parts can be achieved by one of a variety of methods, which range from traditional computer vision techniques to advanced machine learning approaches. Traditional methods often use background subtraction and contour analysis to track human figures in video sequences [44].

Machine learning-based approaches for human body tracking have gained prominence. By leveraging algorithms capable of learning intricate patterns from labeled datasets, they are capable of accurately tracking the human body. Recently, CNNs have emerged as a cutting-edge solution for human body tracking [45,46]. The convolutional layers in CNNs allow for the extraction of meaningful features from images or video frames, enabling robust and accurate human body tracking in diverse scenarios.

3. Implementation

3.1. Assembly and Layout Design

A Universal Robots® UR5e cobot (Odense, Denmark) was used as the main platform. For this purpose, it was necessary to design a layout where the cobot is able to interact with the operator and with parts of a proposed assembly. The layout was designed considering the workspace of an 850 mm radius around the cobot. Figure 2a depicts the computational model of the layout, while Figure 2b shows the experimental layout employed. Additionally, the delimited regions for cobot operation, object storage, object classification, human assembly, and human–robot interaction can be observed. Within the inventory, each object has its own storage area.

The operator is asked to complete a specific task in collaboration with the robot. The task consists of the assembly and disassembly of a connecting rod–crank mechanism. The user can request the system to bring or store the parts in the inventory. All these parts are shown in Figure 3, along with the stock for each part and the full assembly. They were all designed in SolidWorks 2022® (Dassault Systems, Vélizy, France) and later manufactured with fused filament fabrication on a FlashForge® creator pro printer (Zhejiang, China) using PLA.

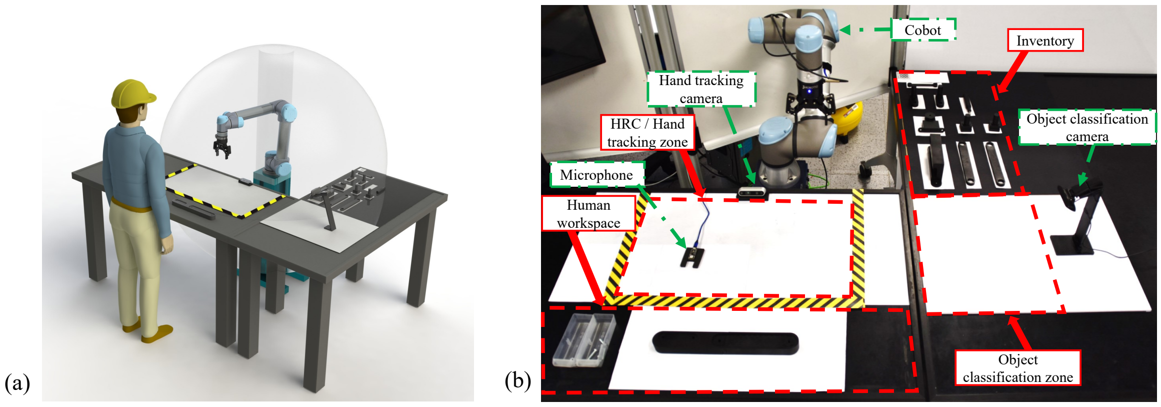


Figure 2. Layout for experimentation. (**a**) CAD isometric view. (**b**) Implemented layout with identified zones (dashed red lines and solid arrows) and components (green, dashed arrows) identified.

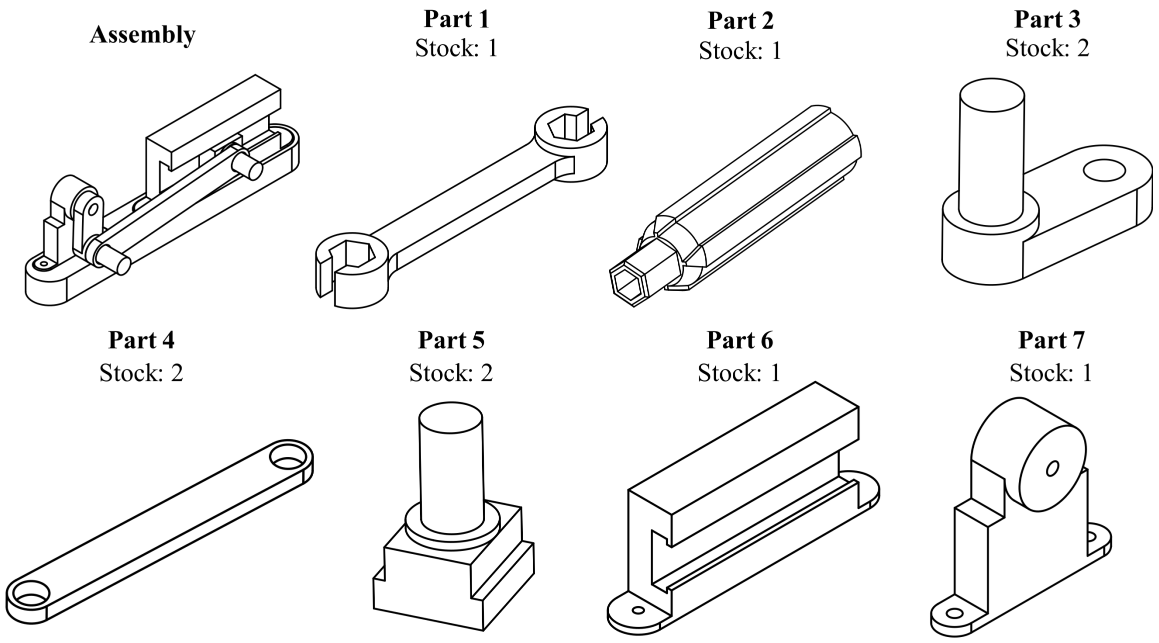


Figure 3. Sketches of printed parts for the assembly of a connecting rod–crank mechanism.

3.2. Robot Operation and Routines

The system is composed of two parallel programs running simultaneously. On the one hand, there is an external computer running the Robot Operating System (ROS) library with several nodes that allow for interaction with various subsystems, which will be described later. On the other hand, within the program running in the control box of the robot, there is a module specifically designed to oversee the waypoints, routines, and calibration. The system obtains the position and orientation of the layout through a marker inspected by a RobotIQ$^{®}$ wrist camera (Lévis, QC, Canada) integrated into the cobot end effector. This combination of programs ensures coordinated control and system operation.

Figure 4 illustrates the various steps and systems involved in each routine. The main loop of the system begins when it expects a "bring" or "take" voice command in the position called "Home". The cobot receives the information from the ROS system through the digital inputs in the control box. Digital signals are sent when voice commands are detected and when objects are classified within the storing sequence (discussed in Sections 3.3 and 3.4).

An inventory control system, present in both the external ROS system and the embedded program within the robot, tracks and manages the total number of parts. Therefore, if the "bring part" routine is executed and no parts are available, the system will provide feedback to the user and request another part, and during the "take part" routine, the system will return the part to the user if the stock is full. Finally, the "return part to user" routine occurs when an object is not recognized by the object classification subsystem and when the detected part has no available space in the inventory. In such cases, the part is returned to the user in the same position from which it was received.

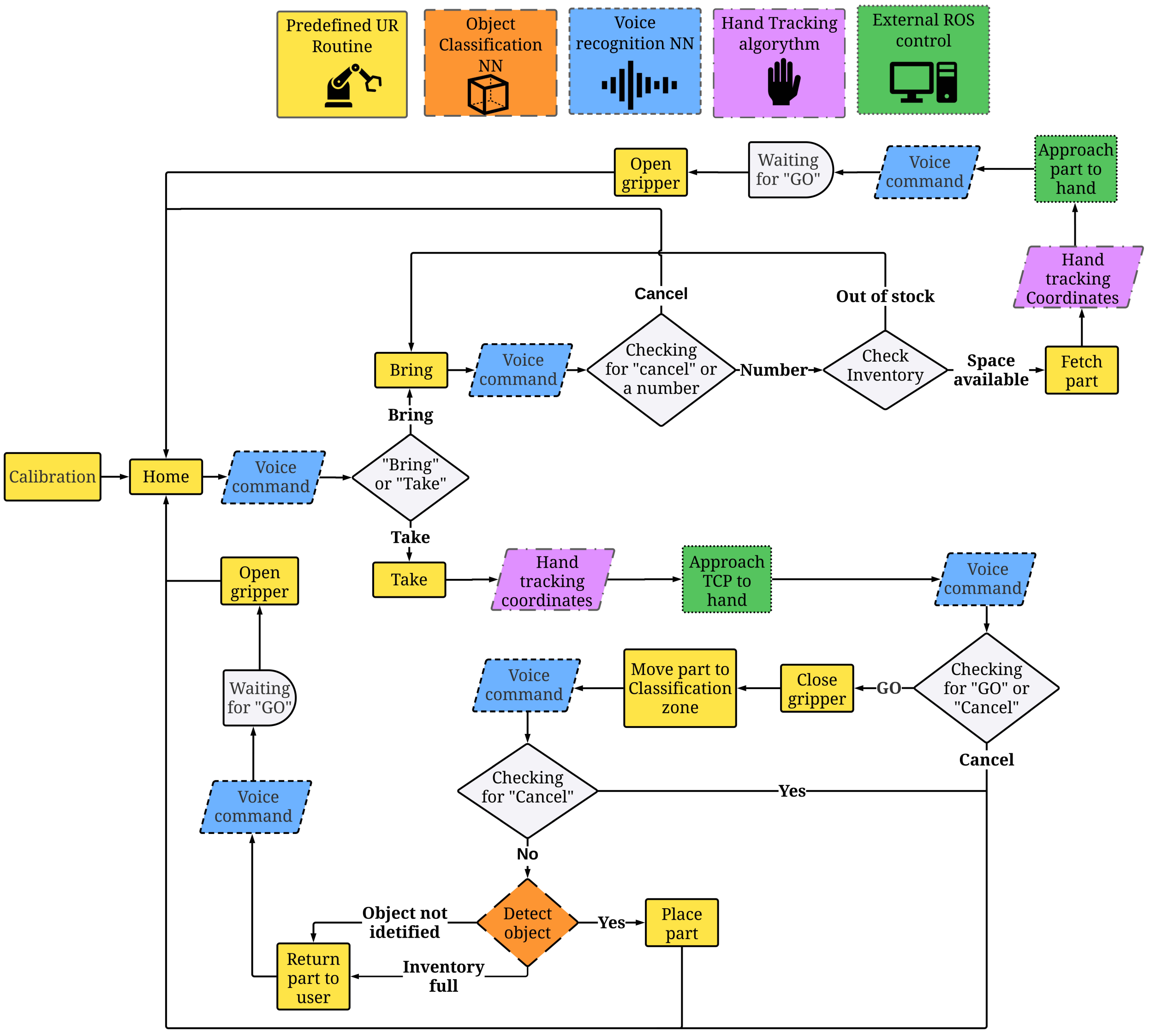


Figure 4. Flow diagram of the robot routines composed of the different subsystems involved.

3.3. Object Classification System

The assembly part classification system enables the effective detection and storage of the various elements of the assembly. To achieve this, the system follows a three-stage process: image acquisition, image preprocessing, and image classification.

A Logitech® C910 camera (Lausanne, Switzerland) was used as the input for the system, as shown in Figure 5. Figure 5a exemplifies the position in which parts were photographed for classification, as well as the setup used for both the training and classification images. Figure 5b shows the robot retrieving the object after it was properly classified. Once the image was captured, data processing was carried out simultaneously in a Python3 script as follows. The images were captured using the OpenCV library. Subsequently, the images were resized to the desired dimensions of 224 $\times$ 224 pixels, which were designated for the design of data pipelines. And finally, contrast and brightness were modified from the captured images. These preprocessing steps allowed us to enhance the differences between objects, enabling the identification of small differences that distinguish the parts.

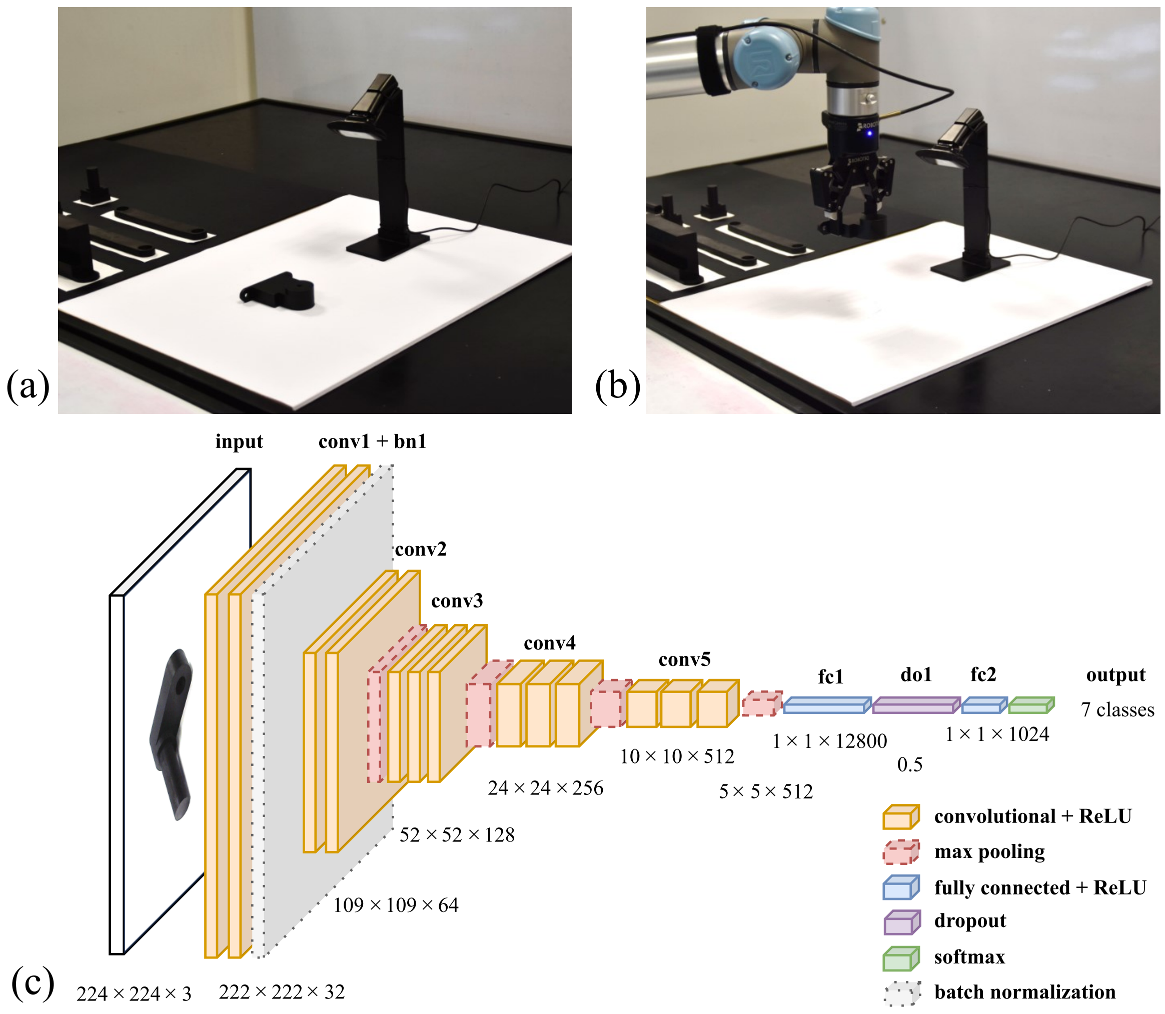


Figure 5. Setup used for object classification. (**a**) The object is positioned for image capture. (**b**) The CNN classifies the object and the cobot retrieves the object for storage. (**c**) Structure and window size of CNN for object classification.

The image classification was performed using a CNN (Convolutional Neural Network) trained with the TensorFlow DL library. The architecture used for the model was a modification of the "AlexNet" first proposed by Krizhevsky et al. [47]. The CNN consists of five convolutional layers, one layer for normalization after the first convolutional layer, max pooling layers after the second to fifth convolutional layers, two fully connected layers, and a drop-out layer.

Figure 5c illustrates the structure of each layer in the CNN, including the window sizes used. To train the CNN, a data pipeline was implemented in Python3 using a dataset obtained from the same setup as the one used for the application.

Each of the seven classes (one for each object observed in Figure 3) was trained with approximately 150 preprocessed images. The dataset was split into 70% used for training, 20% for validation, and 10% for testing. The trained model achieved an accuracy of 0.98 on the test data. The system was integrated into a Python3 script using an ROS node structure. This node subscribed to the camera topic to receive RGB images and published the observed class. The images were classified in 10 consecutive instances within the central node to achieve a precise classification. The mode of these measurements was then used as the final class. This approach ensures accurate classification and prevents the capture of undesired frames.

3.4. Voice Recognition System

In the voice recognition system, a neural network was implemented to classify the audio input from a microphone. The audio recognition process comprises three stages: audio recording, preprocessing, and classification. For audio recording, an nRF52840 SoC Nordic semiconductor® (Trondheim, Norway) processor coupled with an omnidirectional digital microphone MP34DT06JTR from STMICROELECTRONICS® (Geneve, Switzerland) was utilized. The audio amplitude of the audio wave is captured in real-time, allowing for the collection of these amplitudes over time to reconstruct and analyze the audio wave at a later time.

The audio wave capture initiates when the amplitude exceeds a predetermined, experimentally set threshold. This ensures that the recording begins when the microphone detects a high level of sound activity. Each audio sample has a duration of 2 s and contains 3327 amplitude registers. As mentioned in Section 2.2, 2D spectrogram representations of audio signals are widely used for audio classification. Therefore, prior to analyzing the audio samples collected, the audio data were processed by a Python3 script that generated a 2D spectrogram.

The spectrogram is generated by applying a Hamming window to the audio segments. A fast Fourier transform with a size of 256 was used, along with a 50% overlap between segments, representing the magnitude of spectral coefficients. This provides a visual representation of the spectral energy of the audio signal over time, enabling classification in the voice recognition system. Once the audio is represented as a 2D spectrogram, the voice recognition system utilizes a CNN to classify the audio into 1 of the 11 different classes recognized as commands. The neural network consists of four convolutional layers, each followed by a max pooling layer and a final dense layer with SoftMax activation to classify audio into the specified commands, as illustrated in Figure 6.

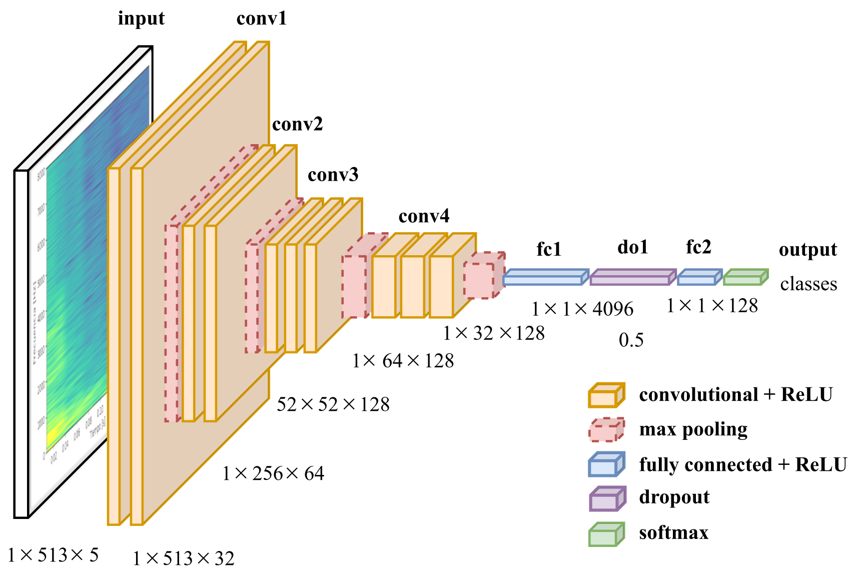


Figure 6. CNN for voice command structure and sizes.

The neural network was trained with a dataset of 300 audio samples per class, recorded by 10 different individuals. An additional noise class was added to recognize when no command was spoken. The dataset was distributed as follows: 80% for training, 10% for validation, and 10% for testing. To simplify the problem, two different models were trained, one to recognize numbers and the other one to recognize actions. Both models included the command 'cancel' to be able to use this command regardless of the model. The whole process was integrated into a Python3 script running an ROS node, which receives the audio waveform as an input and outputs the class predicted by a neural network and the confidence rate of the prediction made. This node also oversees switching between the two models (numbers and actions models) used along the robot routines.

3.5. Hand-Tracking System

The objective of the hand-tracking subsystem is to determine the location of the hand in space and facilitate human–machine interaction for part delivery and retrieval within the assembly space. The Metacarpophalangeal Joint (MCP) in the middle finger was used as the reference point of the hand (Figure 7a). The system consists of three stages: hand capture, hand detection processing, and coordinate array generation.

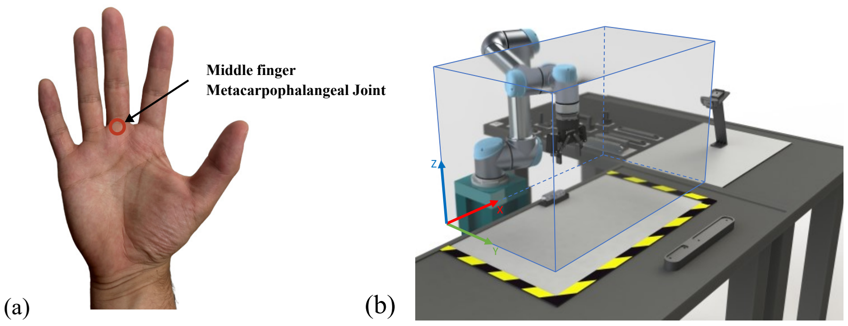


Figure 7. (**a**) The Metacarpophalangeal Joint (MCP), located in the hand of the user, can be located to allow for hand tracking. (**b**) Robot–Human collaboration/hand-tracking zone.

A volume was defined inside the workspace of the robot where human–robot collaboration was permitted, as shown in Figure 7b. An Intel® RealSense Depth Camera D435 (Santa Clara, CA, USA) was necessary to accurately capture the depth, ensuring the end effector could reach any desired point. Hand recognition was performed using the MediaPipe library, which employs a pre-trained model for real-time image processing and the extraction of information about the hand position of the operator. The MediaPipe process consists of two stages: The first one receives the whole RGB image as an input and locates hands with oriented bounding boxes. Then, another model is applied to the hand bounding box region and outputs a matrix with landmarks for the detected palm. One of these landmarks is located at the Metacarpophalangeal Joint, which is the target of the hand recognition system [48].

The robot is commanded to move to the position of the MCP joint as long as it is detected within the allowed volume. It uses the Inverse Geometric Model (IGM) of the robot to calculate the joint-based movements. Figure 8 exemplifies the application of the hand-tracking subsystem, in Figure 8a for bringing a part and in Figure 8b for taking and storing the part.

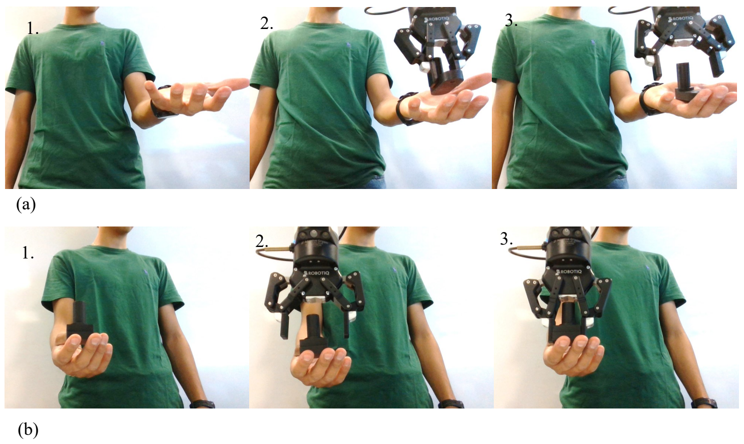


Figure 8. Use example of hand tracking. (**a**) The robot brings a part to the user for assembly. (**b**) The user gives a part to the cobot for storage.

3.6. Systems Integration

All subsystems in the project communicate with a Central Processing System (CPS) through an ROS node. The CPS acts as an orchestrator, synchronizing the activation and communication between the subsystems mentioned earlier. It receives the output of each subsystem and temporarily stores it until another system requires the feedback.

Figure 9 presents the CPS node, designed to receive three inputs: hand position coordinates, recognized voice commands, and the class of the detected object. The CPS node processes these data and transmits signals to the cobot, enabling control over its inner program routines. To facilitate precise control of the movements performed by the cobot, the UR ROS Driver library was used to guide the end effector toward the desired hand position. UR ROS Driver is an ROS library used to control a Universal Robots® cobot; this library contains communication functions that allow one to send movement commands via ROS topics to the cobot. For signal transmission, digital signals are sent to the cobot via digital inputs.

Additionally, each subsystem sends status messages to the CPS, providing valuable information for monitoring the overall state and performance of the system. The CPS also generates its status messages, which help the user interact with the robot. To enhance user-friendliness, these messages undergo audio conversion using the gTTS (Google Text-to-Speech) Python3 library and are subsequently played back, enabling users to effortlessly communicate and control the system through speech and auditory feedback.

The audio feedback provided by the system includes messages indicating when and which commands the robot can receive, the detected object, and alerts for any issues that arise, such as insufficient storage space for an additional piece or when an object is detected unexpectedly.

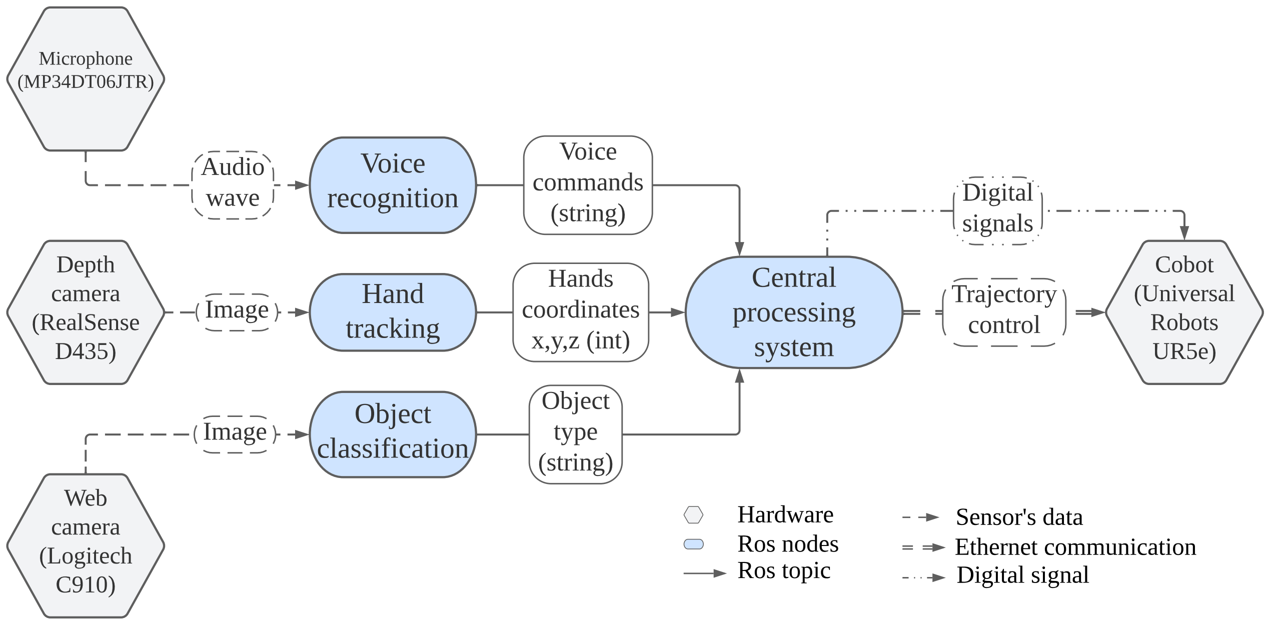


Figure 9. CPS intercommunication with the different systems.

4. Experimental Setup: Isolated Systems and Integration

For the object classification system evaluation, a total of 30 tests were performed for each of the seven parts, resulting in an accumulated 210 tests. These tests were carried out using a fixed setup and constant lighting. To assess the accuracy of the classification system, prediction and confidence rates were recorded for each test.

The voice recognition system was tested with the assistance of 20 volunteers. A total of 200 tests per command were carried out to determine the accuracy of the system in a controlled environment. Each volunteer was responsible for pronouncing commands from the generated model; this consisted of seven commands for numbers (“one”, “two”, "three", "four", "five", "six", "seven") and four commands for instructions ("bring", "take", "go", "cancel"). The command "cancel" was tested 400 times. Of the 20 participants, 7 were women and 13 were male.

Voice command recognition tests were performed inside a closed and noiseless space. The audio signals were sent to a computer running the CNN model and a program that evaluated the performance of each prediction. During the tests, each volunteer sat in a chair positioned 25 cm away from a table where the microphone was connected to the computer. Once the program was running, the detected command and its corresponding confidence rate were recorded.

Hand-tracking subsystem tests were performed by positioning a hand 50 cm parallel to the Intel® RealSense depth camera while running the hand-tracking program. The cobot end effector moved to the predicted position of the hand and the distance between the MCP and the end effector was measured.

The success of the tests was evaluated by assessing the accuracy of the predicted hand position. A predetermined success threshold of 50 mm was set, indicating that if the predicted position of the hand fell within a 50 mm range of the actual position, the test was deemed successful. The tests were performed by 10 different users, and measurements were captured after each run.

For testing the integrated system, a total of 560 routines were performed by 16 individuals. Each person had to construct the assembly twice by performing a complete "bring part" and "take part" routine. Figure 10a exemplifies the "bring part" routine, and Figure 10b exemplifies the take part routine for object number 5. These tests were conducted in a closed and dedicated room to minimize external noise. During the tests, the individual carrying them out stood in front of the table designated for human–robot interaction, positioned in front of the Intel® depth camera, and 25 cm away from the microphone. To evaluate these tests, all the results and errors obtained with the system were recorded. For a test to be considered successful, it had to fulfill the following criteria: correct voice command recognition, accurate hand tracking (distance to MCP under the 50 mm threshold), correct object classification, execution of adequate UR5e routines, and real-time inventory updates for each part in both systems.

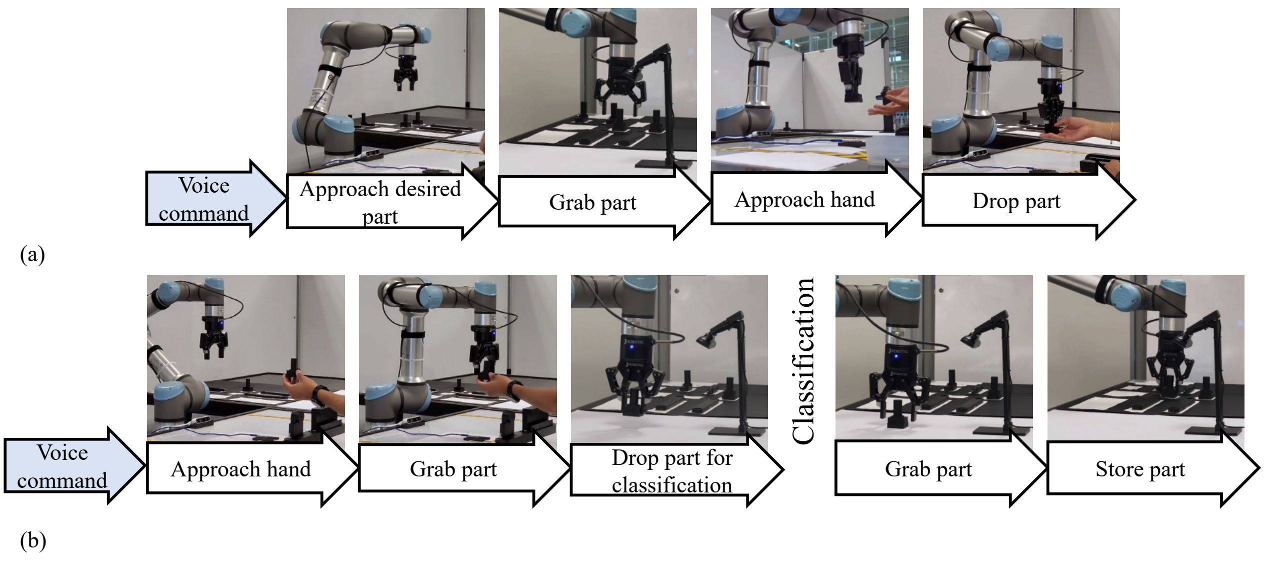


Figure 10. Visual examples of routines. (**a**) "Bring part" routine. (**b**) "Take (and store) part" routine.

5. Results

5.1. Object Classification

The object classification system achieved the results observed in Figure 11, where the confidence rates of all predictions made by the image classification CNN across all tests are shown. From the 210 tests, the average confidence rate obtained was 98.42%, with classes 5 and 6 exhibiting the highest number of outliers. However, all the tests performed yielded average results within the predefined threshold of 95%, considering that in all cases the classes were adequately predicted. It should be noted that while the tests were conducted under the expected conditions of the position and lighting within the system, variations in these aspects may potentially result in incorrect object classifications.

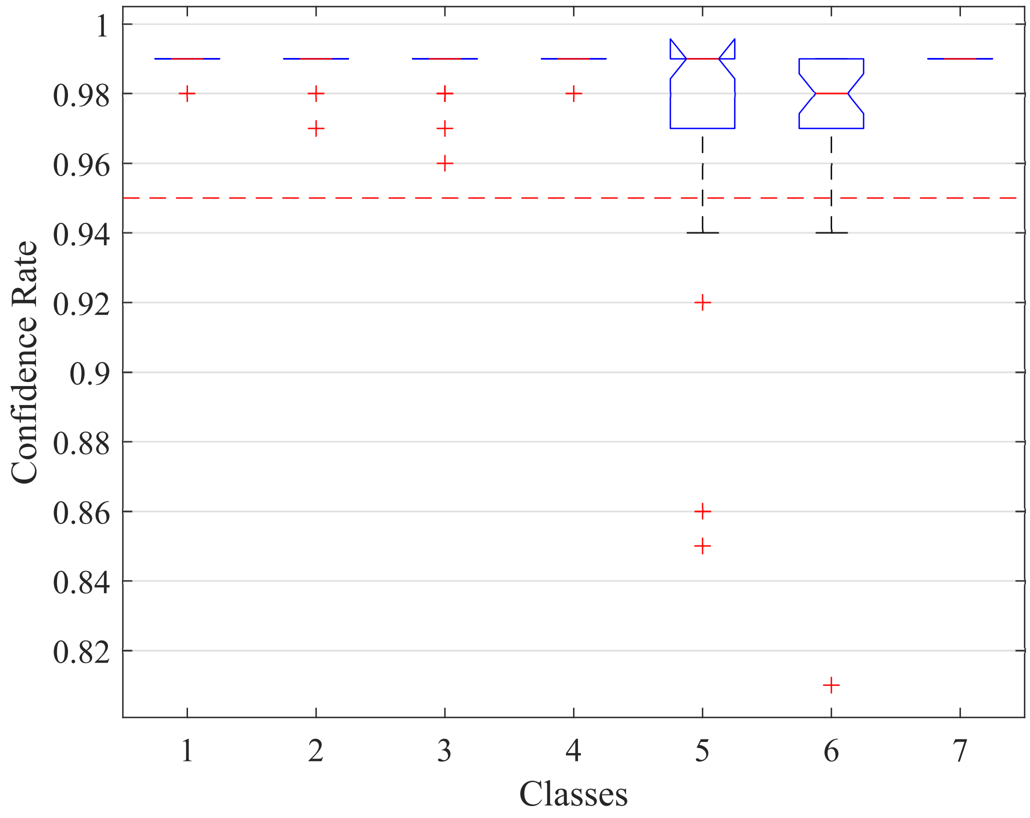
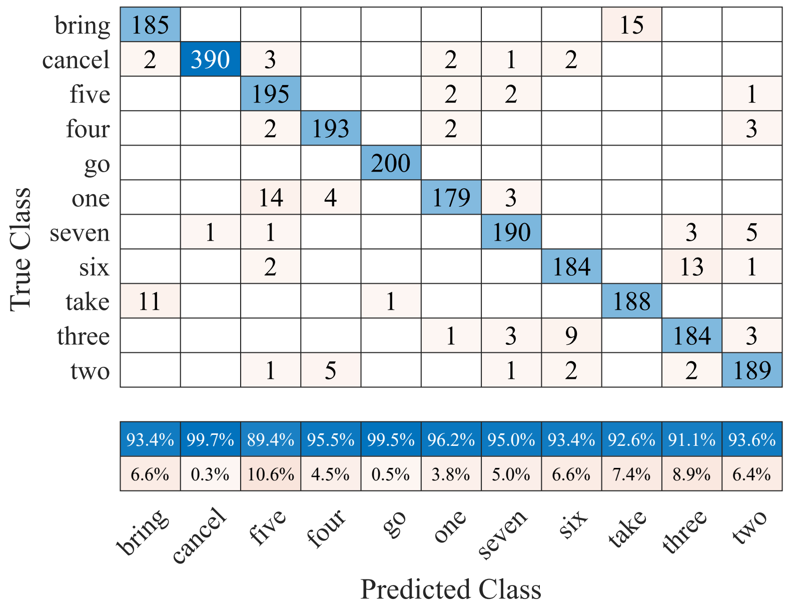


Figure 11. Box plot showing the confidence rate from the CNN classification of the objects within the 210 object classification system tests performed.

5.2. Voice Recognition

The voice recognition system obtained an average accuracy of 94.5% from the 200 tests conducted. Figure 12 displays the obtained predictions. The command "five" had an 89.4% accuracy; in most of the mistaken predictions, the command was confused with "one". The prediction mistakes are attributed to the low sample rate used for data acquisition and command recording, variations in word pronunciation, and voice volume. It was observed that better results were obtained under isolated conditions. The system presented a high sensitivity to background noise and better performance to low-pitched voices. This is attributed to the training dataset used.

Misclassification of voice commands may lead to the execution of incorrect routines. However, the "cancel" command serves to halt misidentified routines, as depicted in Figure 4. Moreover, our tests revealed that the "cancel" command demonstrated a prediction accuracy of 99.7%, as observed in Figure 12. For safety measures, an emergency stop button is available for immediate use when necessary.



True Class \ Predicted Class	bring	cancel	five	four	go	one	seven	six	take	three	two
bring	185								15		
cancel	2	390	3			2	1	2			
five			195			2	2				1
four			2	193		2					3
go					200						
one			14	4		179	3				
seven		1	1				190			3	5
six			2					184		13	1
take	11				1				188		
three						1	3	9		184	3
two			1	5			1	2		2	189
	93.4%	99.7%	89.4%	95.5%	99.5%	96.2%	95.0%	93.4%	92.6%	91.1%	93.6%
	6.6%	0.3%	10.6%	4.5%	0.5%	3.8%	5.0%	6.6%	7.4%	8.9%	6.4%

Figure 12. Confusion matrix showing the results from the voice recognition system tests.

5.3. Hand Tracking

In the hand-tracking subsystem, the distance measurements were made from the end effector to the MCP joint of the user, presenting a 99% effectiveness from the 100 isolated tests. It was observed that users adapted to the motion of the robot and approach seamlessly, performing the tests adequately. The interaction area was marked to indicate the human workspace and the hand-tracking zone as seen in Figure 2b. This helped the users to position their hands within the collaboration area. Most measurements ranged between 10 and 30 mm, falling underneath the 50 mm threshold used to determine the success of each test.

5.4. Integrated System

The integrated system tests achieved an average accuracy of 91.25%, with 49 out of 560 full routines failing. Most errors occurred in the voice command recognition system (47 failures) due to the misclassification of commands by the CNN. Hand recognition had two failures when exceeding the 50 mm distance limit. Object classification, robot routines, and inventory checks worked adequately in all tests. While user experience with cobots and gender did not impact results significantly, high-pitched voices were found to originate more classification errors in voice recognition.

Based on the performed experimentation, the optimal performance of the neural networks requires a controlled environment. The audio recognition system faced challenges with background noise, and the object classification CNN performed well under controlled conditions but can be affected by lighting, background, and object orientation. To enhance the voice recognition performance of the CNN, a more diverse dataset encompassing pronunciation, tone, and voice volume variations is recommended.

The hand-tracking pre-trained model performed accurately. However, external control resulted in longer trajectory calculations. To improve stability and robot movement, transferring complete routine control to internal control is suggested by migrating the controller responsible for hand position from the ROS driver to the internal control of the cobot. These enhancements would lead to an improved response time.

6. Discussion

CNN models, such as the ones implemented here, offer a significant advantage for human–robot collaborative tasks due to their adaptability based on training. Compared to traditional vision-based models, they require relatively shorter training times while swiftly adapting to new sets of parts. While other architectures, such as ResNet or MobileNet, could have been implemented, the model selection was influenced by its shorter training duration and controlled experimental conditions. The CNN models implemented here exhibited high accuracy within a short training period, making them suitable for the current application. However, different conditions, such as varying lighting or movement scenarios, might warrant the utilization of more robust models. A similar discussion could be presented for voice recognition models such as Recurrent Neural Networks, which have shown excellent performance [49]. Once again, a CNN proved sufficient for discerning 11 commands. More robust models could be advantageous for applications involving larger command sets.

The proposed integrated system reached an accuracy of 91.25%. Individually, the object classification model achieved 98.42% accuracy, voice command recognition averaged 94.5%, and hand tracking reached 99% accuracy. The high accuracy of the object classification system is attributed to the robustness of the model in classifying the objects in the lighting conditions used. In the case of the integrated system, the accuracy was affected by errors from the different systems, mostly by voice command misclassification. These results are in line with other previously reported works for both the integrated system and each independent model.

For instance, Murali et al. [15] developed a human–robot collaboration system based on AND/OR graphs, achieving an 80% success rate in 30 tests. On the other hand, Paxton et al. [50] proposed a behavioral tree-based task planning system that reported 100% accuracy in 10 tests across different tasks. However, this system lacked active human interaction beyond a graphical interface. Both aforementioned systems represent human–robot collaborations without the integration of machine learning algorithms.

In contrast, Liu et al. [33] implemented various neural networks in a human–cobot environment, achieving 70% accuracy for body posture estimation, 96% for hand motion recognition, and 93% for voice command recognition. Yet, these models were not deployed for human–robot task completion. Further developments include the work by Rodrigues et al. [32], which presents a CNN model for detecting collisions between users and cobots with an accuracy of 89%, and by Morrison et al. [51], which develops a generative grasping CNN-based model achieving between 88% and 94% accuracy for grasping static and moving objects. Additionally, Sekkat et al. [29] utilized a reinforcement learning model for grasping tasks with a 5-DOF robot, reporting 95.5% accuracy, albeit without human interaction. In summary, current accuracy benchmarks for human–robot collaboration tasks surpass 80%. CNN models in collaborative robot applications exhibit between 70% and 95% accuracy.

Furthermore, the conducted experimentation surpassed the existing literature by encompassing 560 full routines for testing. The experimentation involved distinct evaluation procedures for the models (outlined in Section 4) and an assessment of integrating various models in a practical use-case scenario, yielding comprehensive results. Supplementing quantitative measurements, qualitative evaluations via questionnaires could enhance the findings derived from the experimentation.

7. Conclusions

In this study, three CNNs were successfully integrated to enable human–cobot interaction in a predefined assembly task. The system utilized voice commands for executing routines, a pre-trained hand-tracking model for part handover and storage, and image classification to ensure the accurate placement of the parts.

The results demonstrated the effectiveness of each CNN, with a success rate of 91.25%. It was observed that previous experience in the manipulation of cobots did not change the results of the tests, pointing to the seamless adaptation and use of the system. These findings highlight the significant contribution of the implemented neural networks in enhancing the capabilities of the cobot in assembly tasks, reducing the need for manual intervention, and moving towards a more human-like human–machine interaction.

The success can be attributed to the accurate interpretation of user commands through the audio command classification system and the object-type identification system, which eliminates the requirement for user intervention in identifying object types. In addition, the use of the hand-tracking system enables flexible human–robot collaboration.

Further work could focus on exploring different Deep Learning architectures or implementing different collaborative human–robot activities. The models performed adequately in the environment used for experimentation; nonetheless, by using more robust models, the system could be used in environments with more variations in lighting or movement for the object classification models, or one could implement a bigger set of voice commands.

The use of Deep Learning has the potential to push collaborative human–robot systems to new frontiers. The integration of several CNNs working together in a collaborative task allows one to broaden the range of capabilities. For instance, with the use of voice commands, users are able to perform activities while waiting for an object to be brought or stored. The use of flexible routines and tasks, as seen in [52,53], would enable adaptable routines according to specific application needs.

The evaluation methodology proposed for individual subsystems and their integration can be further utilized in similar projects. The integration of Deep Learning and cobots is widely employed across various domains, ranging from manufacturing to medical assistance. Furthermore, this work serves as a proof of concept for seamlessly integrating multiple CNNs in a collaborative system.

Author Contributions: Conceptualization, E.M., O.O., V.H.S.-H. and D.O.-G.; methodology, E.M. and O.O.; software, E.M., O.O., J.A.L.-S., V.H.S.-H. and D.O.-G.; validation, O.O., J.A.L.-S. and V.H.S.-H.; formal analysis, E.M., O.O., C.L.-D. and I.N.A.-G.; investigation, E.M. and O.O.; writing—original draft preparation, E.M. and O.O.; writing—review and editing, E.M., O.O., M.A.P., I.N.A.-G. and A.G.; data curation, E.M., O.O., C.L.-D. and J.A.L.-S.; visualization, E.M., O.O., C.L.-D. and D.O.-G.; supervision and project administration, E.M., O.O., C.L.-D., E.L.-d.-R. and A.G. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Tecnologico de Monterrey, Vicerrectory of Research and Technology Transfer.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to their large volume.

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

References

1. Galin, R.; Meshcheryakov, R. Review on human–robot interaction during collaboration in a shared workspace. *Lect. Notes Comput. Sci.* **2019**, *11659*, 63–74. [CrossRef]
2. Tsarouchi, P.; Michalos, G.; Makris, S.; Athanasatos, T.; Dimoulas, K.; Chryssolouris, G. On a human–robot workplace design and task allocation system. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 1272–1279. [CrossRef]
3. Malik, A.A.; Pandey, V. Drive the Cobots Aright: Guidelines for Industrial Application of Cobots. In Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, St. Louis, MO, USA, 14–17 August 2022; Volume 5. [CrossRef]
4. Müller, R.; Vette, M.; Mailahn, O. Process-oriented Task Assignment for Assembly Processes with Human-robot Interaction. *Procedia CIRP* **2016**, *44*, 210–215. [CrossRef]
5. El Zaatari, S.; Marei, M.; Li, W.; Usman, Z. Cobot programming for collaborative industrial tasks: An overview. *Robot. Auton. Syst.* **2019**, *116*, 162–180. [CrossRef]
6. Cherubini, A.; Passama, R.; Crosnier, A.; Lasnier, A.; Fraisse, P. Collaborative manufacturing with physical human–robot interaction. *Robot. Comput.-Integr. Manuf.* **2016**, *40*, 1–13. [CrossRef]
7. Galin, R.; Meshcheryakov, R.; Kamesheva, S.; Samoshina, A. Cobots and the benefits of their implementation in intelligent manufacturing. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *862*, 032075. [CrossRef]
8. Javaid, M.; Haleem, A.; Singh, R.P.; Rab, S.; Suman, R. Significant applications of Cobots in the field of manufacturing. *Cogn. Robot.* **2022**, *2*, 222–233. [CrossRef]
9. Shaikh, T.A.; Rasool, T.; Verma, P. Machine intelligence and medical cyber-physical system architectures for smart healthcare: Taxonomy, challenges, opportunities, and possible solutions. *Artif. Intell. Med.* **2023**, *146*, 102692. [CrossRef]

10. Shinde, P.P.; Shah, S. A Review of Machine Learning and Deep Learning Applications. In Proceedings of the 2018 4th International Conference on Computing, Communication Control and Automation (ICCUBEA 2018), Pune, India, 16–18 August 2018. [CrossRef]
11. Rai, R.; Tiwari, M.K.; Ivanov, D.; Dolgui, A. Machine learning in manufacturing and industry 4.0 applications. *Int. J. Prod. Res.* **2021**, *59*, 4773–4778. [CrossRef]
12. Borboni, A.; Reddy, K.V.V.; Elamvazuthi, I.; AL-Quraishi, M.S.; Natarajan, E.; Ali, S.S.A. The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works. *Machines* **2023**, *11*, 111. [CrossRef]
13. Semeraro, F.; Griffiths, A.; Cangelosi, A. Human–robot collaboration and machine learning: A systematic review of recent research. *Robot. Comput.-Integr. Manuf.* **2023**, *79*, 102432. [CrossRef]
14. Makrini, I.E.; Merckaert, K.; Lefeber, D.; Vanderborght, B. Design of a collaborative architecture for human-robot assembly tasks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1624–1629. [CrossRef]
15. Murali, P.K.; Darvish, K.; Mastrogiovanni, F. Deployment and evaluation of a flexible human–robot collaboration model based on AND/OR graphs in a manufacturing environment. *Intell. Serv. Robot.* **2020**, *13*, 439–457. [CrossRef]
16. Chen, M.; Soh, H.; Hsu, D.; Nikolaidis, S.; Srinivasa, S.; Chen, M.; Nikolaidis, S.; Soh, H.; Hsu, D. Trust-Aware Decision Making for Human-Robot Collaboration. *ACM Trans. Hum.-Robot Interact. (THRI)* **2020**, *9*, 1–23. [CrossRef]
17. Shukla, D.; Erkent, O.; Piater, J. Learning semantics of gestural instructions for human-robot collaboration. *Front. Neurorobotics* **2018**, *12*, 7. [CrossRef] [PubMed]
18. Rozo, L.; Silvério, J.; Calinon, S.; Caldwell, D.G. Learning controllers for reactive and proactive behaviors in human-robot collaboration. *Front. Robot. AI* **2016**, *3*, 30. [CrossRef]
19. Munzer, T.; Toussaint, M.; Lopes, M. Efficient behavior learning in human–robot collaboration. *Auton. Robot.* **2018**, *42*, 1103–1115. [CrossRef]
20. Grigore, E.C.; Roncone, A.; Mangin, O.; Scassellati, B. Preference-Based Assistance Prediction for Human-Robot Collaboration Tasks. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4441–4448. [CrossRef]
21. Nikolaidis, S.; Ramakrishnan, R.; Gu, K.; Shah, J. Efficient Model Learning from Joint-Action Demonstrations for Human-Robot Collaborative Tasks. In Proceedings of the 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Portland, OR, USA, 2–5 March 2015; pp. 189–196. [CrossRef]
22. Chen, X.; Jiang, Y.; Yang, C. Stiffness Estimation and Intention Detection for Human-Robot Collaboration. In Proceedings of the 15th IEEE Conference on Industrial Electronics and Applications (ICIEA 2020), Kristiansand, Norway, 9–13 November 2020; pp. 1802–1807. [CrossRef]
23. Yip, M.; Salcudean, S.; Goldberg, K.; Althoefer, K.; Menciassi, A.; Opfermann, J.D.; Krieger, A.; Swaminathan, K.; Walsh, C.J.; Huang, H.H.; et al. Artificial intelligence meets medical robotics. *Science* **2023**, *381*, 141–146. [CrossRef]
24. Droste, R.; Drukker, L.; Papageorghiou, A.T.; Noble, J.A. Automatic Probe Movement Guidance for Freehand Obstetric Ultrasound. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2020: 23rd International Conference, Lima, Peru, 4–8 October 2020; Proceedings, Part III; Springer: Berlin/Heidelberg, Germany, 2020; pp. 583–592. [CrossRef]
25. Ahmad, M.A.; Ourak, M.; Gruijthuijsen, C.; Deprest, J.; Vercauteren, T.; Poorten, E.V. Deep learning-based monocular placental pose estimation: Towards collaborative robotics in fetoscopy. *Int. J. Comput. Assist. Radiol. Surg.* **2020**, *15*, 1561–1571. [CrossRef]
26. Kwon, J.; Park, D. Hardware/Software Co-Design for TinyML Voice-Recognition Application on Resource Frugal Edge Devices. *Appl. Sci.* **2021**, *11*, 11073. [CrossRef]
27. Ionescu, T.B.; Schlund, S. Programming cobots by voice: A human-centered, web-based approach. *Procedia CIRP* **2021**, *97*, 123–129. [CrossRef]
28. Matsusaka, Y.; Fujii, H.; Okano, T.; Hara, I. Health exercise demonstration robot TAIZO and effects of using voice command in robot-human collaborative demonstration. In Proceedings of the RO-MAN 2009—The 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, 27 September–2 October 2009; pp. 472–477. [CrossRef]
29. Sekkat, H.; Tigani, S.; Saadane, R.; Chehri, A.; García, O.R. Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping. *Appl. Sci.* **2021**, *11*, 7917. [CrossRef]
30. Gomes, N.M.; Martins, F.N.; Lima, J.; Wörtche, H. Reinforcement Learning for Collaborative Robots Pick-and-Place Applications: A Case Study. *Automation* **2022**, *3*, 223–241. [CrossRef]
31. Aswad, F.E.; Djogdom, G.V.T.; Otis, M.J.; Ayena, J.C.; Meziane, R. Image generation for 2D-CNN using time-series signal features from foot gesture applied to select cobot operating mode. *Sensors* **2021**, *21*, 5743. [CrossRef] [PubMed]
32. Rodrigues, I.R.; Barbosa, G.; Filho, A.O.; Cani, C.; Sadok, D.H.; Kelner, J.; Souza, R.; Marquezini, M.V.; Lins, S. A New Mechanism for Collision Detection in Human–Robot Collaboration using Deep Learning Techniques. *J. Control. Autom. Electr. Syst.* **2022**, *33*, 406–418. [CrossRef]
33. Liu, H.; Fang, T.; Zhou, T.; Wang, Y.; Wang, L. Deep Learning-based Multimodal Control Interface for Human-Robot Collaboration. *Procedia CIRP* **2018**, *72*, 3–8. [CrossRef]

34. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [CrossRef] [PubMed]
35. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905. [CrossRef]
36. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]
37. Chen, Q.; Song, Z.; Dong, J.; Huang, Z.; Hua, Y.; Yan, S. Contextualizing object detection and classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 13–27. [CrossRef]
38. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
39. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
40. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. In Proceedings of the Computer Vision Conference (CVC 2019), Las Vegas, NV, USA, 2–3 May 2019; pp. 128–144. [CrossRef]
41. Singh, A.P.; Nath, R.; Kumar, S. A Survey: Speech Recognition Approaches and Techniques. In Proceedings of the 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, 2–4 November 2018; pp. 1–4. [CrossRef]
42. Lyashenko, V.; Laariedh, F.; Sotnik, S.; Ayaz Ahmad, M. Recognition of Voice Commands Based on Neural Network. *TEM J.* **2021**, *10*, 583–591. [CrossRef]
43. Ansari, M.I.; Hasan, T.; Member, S. SpectNet: End-to-End Audio Signal Classification Using Learnable Spectrograms. *arXiv* **2022**, arXiv:2211.09352.
44. Bobick, A.; Davis, J. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [CrossRef]
45. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef] [PubMed]
46. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 483–499.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
48. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv* **2020**, arXiv:2006.10214.
49. Hung, P.D.; Minh, T.; Hoang, L.; Minh, P. Vietnamese speech command recognition using recurrent neural networks. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 194–201. [CrossRef]
50. Paxton, C.; Hundt, A.; Jonathan, F.; Guerin, K.; Hager, G.D. CoSTAR: Instructing collaborative robots with behavior trees and vision. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017. [CrossRef]
51. Morrison, D.; Corke, P.; Leitner, J. Learning robust, real-time, reactive robotic grasping. *Int. J. Rob. Res.* **2020**, *39*, 183–201. [CrossRef]
52. Darvish, K.; Bruno, B.; Simetti, E.; Mastrogiovanni, F.; Casalino, G. Interleaved Online Task Planning, Simulation, Task Allocation and Motion Control for Flexible Human-Robot Cooperation. In Proceedings of the RO-MAN 2018—27th IEEE International Symposium on Robot and Human Interactive Communication, Nanjing, China, 27–31 August 2018; pp. 58–65. [CrossRef]
53. Toussaint, M.; Munzer, T.; Mollard, Y.; Wu, L.Y.; Vien, N.A.; Lopes, M. Relational activity processes for modeling concurrent cooperation. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5505–5511. [CrossRef]