Learning-Finding-Giving: A Natural Vision-Speech-based Approach for Robots to Assist Humans in Human-Robot Collaborative Manufacturing Contexts

Emilio Herrera
School of Computing
Montclair State University
Montclair, USA
herrerae7@montclair.edu

Maxim Lyons
School of Computing
Montclair State University
Montclair, USA
lyonsm1@montclair.edu

Jesse Parron
School of Computing
Montclair State University
Montclair, USA
parronj1@montclair.edu

Rui Li
School of Computing
Montclair State University
Montclair, USA
liru@montclair.edu

Michelle Zhu
School of Computing
Montclair State University
Montclair, USA
zhumi@montclair.edu

Weitian Wang*
School of Computing
Montclair State University
Montclair, USA
wangw@montclair.edu

***Abstract*—Human-robot collaboration can improve and enhance current manufacturing processes, in which robots are able to provide collaborative assistance to humans, allowing for increased productivity and minimal time waste. The use of everyday mechanical appliances and tools is unavoidable, making these handover tasks common in human-robot collaborative manufacturing contexts. A typical handover task can be performed in three general steps: object identification, object grasping, and object handover. In this work, we propose a learning-finding-giving framework based on computer vision and speech recognition approaches for robots to dynamically identify and deliver tools for human partners in collaborative tasks. The YOLOv5 object detection algorithm is utilized for the identification of common mechanical tools. To teach robots to understand the target objects, a custom dataset is created from over 2000 images of the mechanical tools, followed by the implementation in real-world human-robot collaborative tasks. Experimental results and evaluations show that the proposed solution allows robots to dynamically understand and grasp tools with high accuracy, effectively assisting in handover tasks for human teammates.**

Keywords—robotics; human-robot collaboration; handover; smart manufacturing; object detection

I. Introduction

Robotics has become increasingly important, and its applications are found in many places. For example, we use robots to help secure areas, navigate terrain, and monitor dangerous environments or situations [1, 2]. We also use robots for assistance in agricultural tasks [3]. In the current manufacturing industry, robots are frequently used for extremely repetitive and monotonous tasks like those found on assembly lines, handling hazardous materials, or performing dangerous processes. Kas stated that robots have been used to inspect dangerous areas that humans are unable to inspect on their own, but also perform dangerous functions such as cutting, sawing, hydro blasting, vacuum sludge, and more; in turn, reducing human risk [4]. However, humans still provide an abundance of critical value in manufacturing, as stated by Kim, "The role of humans is not reduced, but robotic technology requires different high-level responsibilities in human-robot interaction (HRI)" [5]. For example, humans still have unmatched problem-solving skills and provide creative and unique solutions compared to robots which are usually tasked with solving the same work under the same set of conditions. An increasingly popular implementation is known as human-robot collaboration, which allows robots and humans to coincide on the same task. This paradigm employs the benefit of precise robot movements, while still allowing human expertise and creativity [6, 7].

Human-robot collaboration is the cooperation and interaction of humans with robot assistants. The collaboration between humans and robots creates new opportunities to enhance safety, optimize production, increase efficiency, improve task quality, and empower worker flexibility [8]. Assigning time-consuming, monotonous, or meticulous tasks to robots can allow human collaborators to focus on tasks that require unique and creative solutions. Humans can also use this extra help to better troubleshoot and quickly identify issues in manufacturing processes. Freeing humans from dangerous tasks can also lower the stress and fear of human collaborators. These benefits all manifest in lower costs and a safer environment for companies and their workers.

Handover tasks in human-robot collaboration are the actions in which robots locate, grasp, and give objects or tools to their human collaborator [9, 10]. This kind of task is widespread in human-robot collaboration. A lot of human time and energy can be saved with a proper implementation of the handover task. Existing studies solved this problem by researching human-human handovers and then using observations from those to better inform human-robot handover tasks [11]. They focused on when and where human-robot handover tasks should take place and how they should proceed. Another study utilized mixed reality techniques via HoloLens to visualize human-robot handovers [12]. This enables handovers to be done with robots that have little limitations on their movement allowing for more fluid and more human-like motions.

With interactions between humans and robots being more common, there has been substantial research to make robot interactions more predictable, efficient, and effective. Existing research in the domain of human-robot handover in collaborative tasks has contributed significantly to solving this intricate challenge. Strabala *et al*. [11] have emphasized the

979-8-3503-1579-0/24/$31.00 ©2024 IEEE

replication of natural human handover tasks, controlling key parameters such as reach distance and timing to enhance the intuitiveness of interactions. Huber *et al*. [13] have delved into the realm of effective joint strategies, evaluating the role of 'biological motion' in handover interactions, thereby improving coordination and collaboration between humans and robots. Wang *et al*. [14] have explored the utilization of partial demonstrations to predict human handover intentions, effectively modeling and understanding human behaviors. Furthermore, Castro *et al.* [15] introduced a machine learning-based approach to adapt to diverse handover scenarios, showcasing the adaptability of robots to varying collaborative environments. Researchers have also addressed safety concerns in this context, with Chan *et al.* [16] focusing on advanced perception systems to detect grip strength, ensuring safety during handover interactions, and reducing the risk of accidents. Additionally, a coordinated approach has been explored by Wang *et al.* [17] for multi-agent handovers as well as presenting a method for training robots to handle diverse human behaviors. Moreover, trust-building mechanisms have been suggested by [18, 19], emphasizing transparent communication and reliable execution of tasks especially in manufacturing contexts. The integration of tactile or visual feedback into handover tasks enhances the robot's grasp and object manipulation abilities [20, 21]. Cakmak *et al.* [22] have proposed a design initiative to facilitate effective communication and coordination between humans and robots during handover tasks by advocating for human preferences for optimal handover configurations. These collective efforts in research serve to make collaborative tasks more effective, safer, and adaptable to different contexts.

However, these approaches treat humans as part of the environment as mentioned in [17]. A problem area has been dynamically identifying the object that is being handed over to the human. The authors of [15] brought up the argument that in an industrial environment, voice commands may be ineffective due to background noise and chatter. Overall, the major issue that these researchers try to tackle is the safety of the human operators as well as the robots in control [23, 24]. In a collaborative workspace, there may be multiple robots, so it is vital for the robots to track human movements to prevent collisions. Research elaborated by [25, 26], discussed pressing issues regarding the ergonomics of collaborative robots in human-centered workspaces, specifying that posture plays a key role as well as the human's manipulation capacity.

In given collaborative contexts, it is vital that machines can use vision-based systems for object recognition for various aspects of handover tasks. The authors of [27] utilized a vision system to allow the robot to be the giver and the receiver in handover tasks by adopting a dynamic path-finding algorithm to find the human hand. The work [28] employed deep learning-based perception modules to provide a safe handover experience for previously unseen objects and those labeled in their database. There also has been research demonstrating how the YOLOv5 model can specifically be implemented for picking apples [29]. In [30], the authors discussed the training procedure, active learning, where the model selected its samples for labeling based on uncertainty.

However, robots learning from demonstrations would require accurate perception abilities [26]. This presents a challenge, especially with smaller items—and that is why they suggest that learning programming, or learning from demonstration, should only be utilized with large objects and small objects may benefit from a customized image processing or machine learning algorithm. An unfortunate limitation is how object detection can be a bottleneck in some research [28], where objects that aren't detected wouldn't be handed over so humans can get impatient. The approach that they wanted was a very natural handover experience, but this can make humans unwilling to collaborate. There is also the issue previously mentioned about the intentions of handover. Some users may not intend to initiate a handover, but they may accidentally say a phrase or keyword that would prompt the robot to proceed with the task anyway.

Motivated by the above issues, in this study, we propose a novel learning-finding-giving framework based on computer vision and speech recognition techniques for robots to dynamically identify and deliver tools for human partners in collaborative tasks. The YOLOv5 object detection algorithm is utilized for the identification of common mechanical tools during the human-robot handover process. The developed natural vision-speech-based approach is validated in real-world human-robot collaborative manufacturing contexts.

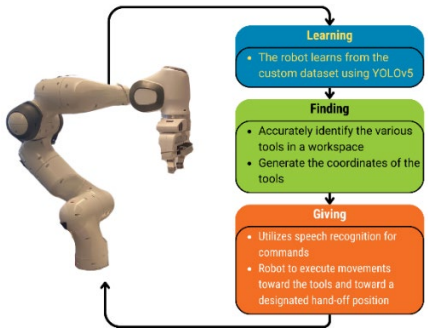

Fig. 1. The proposed learning-finding-giving framework.

II. Approach Overview

The overarching vision of this study is to improve human-robot collaboration quality by incorporating artificial intelligence, computer vision, speech recognition, and eye-in-hand calibration techniques [31]. The proper implementation of these approaches allows for a natural, safe, and effective human-robot collaboration process in different assembly, disassembly, and repair tasks. As shown in Fig. 1, the proposed framework includes three core sections: *Learning, Finding,* and *Giving*. *Learning* will be incorporated with a YOLOv5 model that will enable the robot to learn from a dataset of common tools from our shared workspace demonstration. *Finding* will allow the robot to utilize the trained YOLOv5 model to distinguish between and accurately identify the various tools in a workspace, while also using eye-in-hand calibration to generate the coordinates of the tools. *Giving* will act upon and utilize eye-in-hand calibration, and speech recognition to have the robot execute movements toward the tools and toward a designated hand-off position when given the proper verbal command. These three parts all depend on

each other to create a safe and effective handover of tools. Using this framework, the robot is able to differentiate and locate different tools, obtain accurate coordinates of the tools, and move precisely and effectively for fluid handoffs. During the collaboration process, the human can direct the robot according to their specific needs in the shared task.

III. Modeling Methodology

A. YOLOv5

YOLO (You Only Look Once) is a real-time detection algorithm utilizing a single neural network to predict and classify images in one evaluation [32, 33]. In this study, we utilize an evolved version of YOLO, known as YOLOv5. YOLOv5 is a state-of-the-art real-time deep learning-based object identification system, which consists of pre-trained models and is compared to previous iterations of YOLO based on Darknet. It is lighter and further optimized allowing for faster detection without sacrificing its high accuracy [33]. Similar to previous iterations of YOLO, it works by applying a single convolutional neural network architecture to the whole image. The image is divided into regions for which bounding boxes and probabilities are predicted. As the model is trained, the detection performance is optimized and frame detection is done using regression, which is a lot less complex than other approaches. In this study, we use YOLOv5 to identify objects in our workspace together with eye-in-hand coordination techniques to identify, grasp, and hand over tools to a human.

B. Data Collection and Processing

In this study, we created a dataset for the purpose of training the YOLOv5 model. The dataset originally consisted of 981 images of eight different common mechanical tools which are: adjustable wrenches, linesman pliers, screwdrivers, regular wrenches, utility knives, long nose pliers, torpedo levels, and wire strippers. These were annotated images with class labels and bounding boxes one by one. After applying augmentations to these images changing the hue, brightness, and blur, and adding cutouts, mosaics, shear, and noise, the dataset expanded to 2359 images. These augmentations increase the diversity and performance of the dataset, especially when there are changes in lighting, contrast, and more. For each image, bounding boxes for each class of objects present in the image were properly annotated for the dataset. All photos were also resized to be 640px by 640px because we run inference at this resolution, and it will provide better accuracy and faster inference as a result of the dataset.

C. Human Speech Instruction Parameterization

The approach utilized in this study is based on a continual speech interaction between the human and the robot, as presented in Fig. 2. The robot begins by prompting the human to what mechanical tool they would like. The robot then begins to listen through the microphone for a response. Once the speech is no longer detected, the speech is then recognized through Google speech recognition, and text is generated. This text is used by the robot to decide what tool to pick up. The program scans the text for tool names and takes the tool name that is recognized first. If no tool name is found in the generated text the system will again prompt for another response through the microphone until a tool name is found. After locating, grasping, and performing a handoff maneuver with the tool requested the robot returns to a rest state and loops the previous steps. The microphone used is connected separately and is placed on the edge of the workspace. When the microphone is opened, it first adjusts for ambient noise to obtain clear readings. The microphone turns sounds into electrical signals which are then analyzed by algorithms to determine the word that best fits the sound recorded.

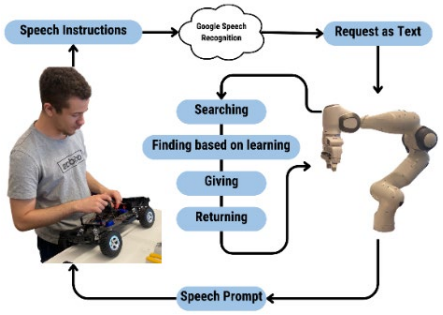

Fig. 2. Diagram of the speech recognition process.

D. Neural Network Configuration

To begin the configuration of our YOLOv5 model, we first chose a pre-trained model's weights 'yolov5s6.pt' from YOLOv5 as our initial weights to train our dataset with. To set up our dataset for training, we utilized Roboflow to create bounding boxes and annotate images for the dataset. The images resized to 640px by 640px match the size we run inference at for training and testing. A data.yaml file was created that specifies the file path to the training, validation, and testing portions of our dataset as well as the number of classes and those class names in the YOLOv5. 88% of the images were put into training, 8% were used for validation and the final 4% were used for testing. Inside each of the training, validation, and testing dataset files there were two subfolders, one containing the images themselves, and another named 'labels' that contained bounding box information for each image. The training parameters are shown in Table I. The training stopped after the epochs we specified were completed which was 100, where 1 epoch is one cycle through the dataset. After the training was completed, two weights files were generated: one for the best mAP@0.5 (mean average precision) score, and another for the final epoch's run through the dataset. The training metrics and figures were also generated to better understand the model's performance.

TABLE I. Parameters and their values used when running the training of our custom YOLOv5 dataset

Parameter	Value	Description
Image	640	Image size to run training at
Batch Size	32	Number of samples to run in one batch
Epochs	100	Number of complete iterations to learn from training examples
Data	data.yaml	Name of a file that specifies the path to the dataset and classes
Weights/Model	yolov5s6.pt	Name of file containing the pre-trained weights/model

Cache	disk	Optional parameter to cache information to either disk or RAM for faster training
Device	0	Optional parameter to specify which GPU/CPU to run training with

E. Vision-Based Object Detection and Picking Up

Attached to the collaborative robot is an Intel D435i RealSense depth camera with which we run YOLOv5's inference with a confidence threshold of 70%. Confidence refers to the score associated with each prediction made by the model, the score is generated from the model's certainty of the accuracy of the predicted class label and the predicted bounding box for the detected object. To detect, locate, and pick up objects, we utilize an eye-in-hand calibration method. To calibrate our RealSense camera, we first obtain extrinsic and intrinsic rotation matrices and translation vectors to apply corrective transformations. Then utilize the hand-eye calibration package in MoveIt! [34], we took photos of an ArUco marker from varying angles and positions and ran an algorithm developed by Daniilidis to determine the correct transformation to apply to convert the pixel coordinates of an object to a real-world object [35]. After applying those transformations, we can obtain the camera pixel coordinates of the center of an object from YOLOv5 inference based on the location of the bounding boxes around detected objects. Then we sent and looked up transforms via a transformation module to receive real-world coordinates for our robot.

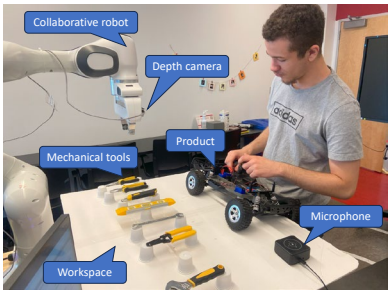

Fig. 3. The experimental platform.

IV. Experimental Setup

A. Experimental Platform

The experimental platform, as shown in Fig. 3, consists of a collaborative robot, an Intel D435i RealSense depth camera, a microphone, mechanical tools, a product to repair/work on, and the workspace. The collaborative robot is a 7-DoF robot arm with a two-finger parallel gripper that is connected to a local controller [36]. The Robot Operating System (ROS) is used for robot system control. ROS is an open-source framework that provides a vast collection of libraries and tools that allow for complex control and applications of robot systems [37]. Alongside ROS we utilize the package MoveIt! [34], which contains many powerful tools for robot path planning. The Speech Recognition package that utilizes Google speech-to-text API allows for voice commands to direct the robot.

B. Task Description

The proposed framework is validated via a real-world human-robot collaborative task, in which we set up a shared working environment where tools and a product are laid out. Once our experiment and algorithms start running, YOLOv5 will begin detection and the eye-in-hand coordination approach will provide real-time position location for the tools in the frame. The human collaborator will be prompted to ask for a tool. The system will then search for the name of a tool in the response and begin the process for finding it. The robot will then scan both sides of the workspace and find the tool requested on one of the sides. The process for grasping the tool will begin where the robot centers itself above the tool, then descends to grasp it and move to a handoff position where the tool will be handed to the human worker.

V. Results and Evaluations

A. Training and Validation Loss Results and Analysis

Fig. 4 presents the loss of the developed model as the training and validation progressed. The loss in YOLOv5 is the combination of class loss, object loss, and bounding box loss. The Y-axis refers to the percentage of the type of loss. The X-axis refers to the epoch number. Box loss refers to how accurate the predicted bounding boxes of our model are compared to the bounding boxes provided from our annotated dataset. Object loss calculates the error in detecting whether an object is present in a grid cell. Class loss is how correct the classifications of all predicted bounding boxes are. During the training, we can see that all three types of loss decrease rapidly approaching values less than 0.02. Validation shows the same for bounding box loss and class loss. However, object loss during validation doesn't consistently decrease as our epoch increases, rather this metric goes up and down ending about 0.0073.

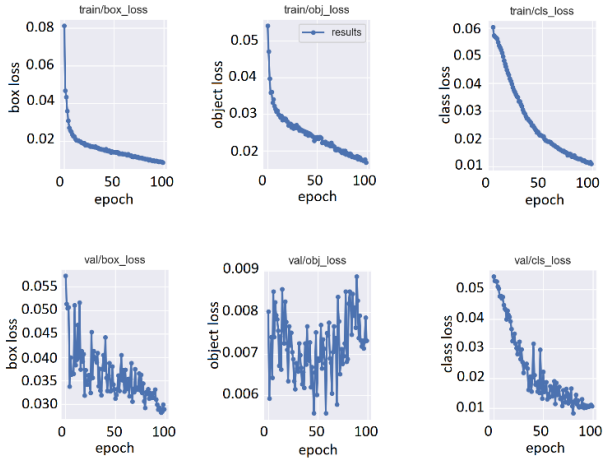

Fig. 4. The training and validation loss.

B. Evaluations

1) Precision-Confidence

The precision-confidence of the developed model is shown in Fig. 5. This metric is a representation of how well our model performs at different confidence thresholds. The results indicate that there is a high degree of precision between confidence levels of about 0.2 to 0.9. They also show that some classes have a drastic drop or rise of precision that dips as low as 0.0 or all the way up to 1.0 when confidence

begins to pass 0.9, which suggests that our model could benefit from an improved and larger dataset. However, between confidence levels of about 0.2 to about 0.8, our model functions well and we have the choice of setting the confidence level low to potentially adjust recall as we require. For the purposes of our experiment, higher confidence was chosen to better phase out potential false positives. All classes hit perfect precision, with some earlier than others. This suggests that all classes could benefit from a larger more diverse set of pictures. Some classes such as the knife or adjustable wrench that hit perfect precision early would benefit the most from this change. Combining this information with the metrics of Precision-Recall helps us determine a more exact confidence threshold to utilize to maximize model performance.

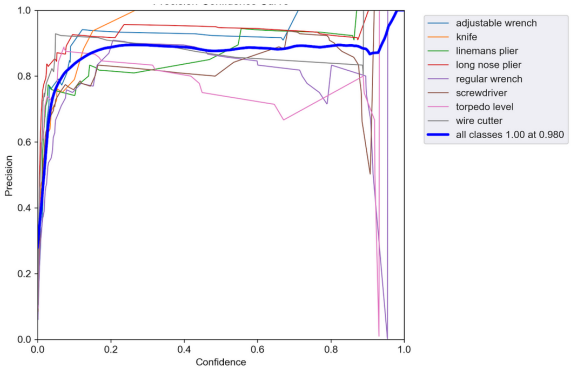

Fig. 5. The precision-confidence of the developed model.

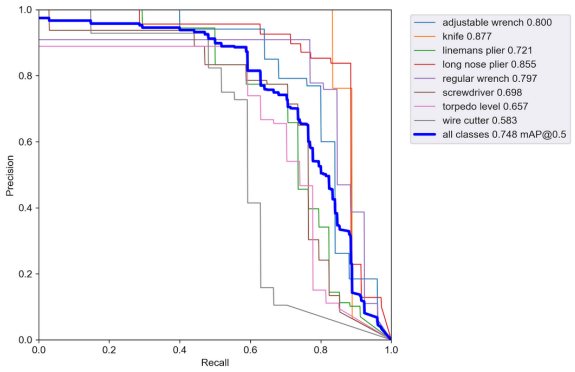

Fig. 6. The precision-recall of the developed model.

2) Precision-Recall

The precision-recall of the developed model in Fig. 6 shows us the performance of our model with the tradeoff between precision and recall. Precision is the accuracy of positive predictions while recall refers to the ability of the model to classify all positive instances. A high area under the curve represents high recall and high precision, where high precision results in a low false positive rate and high recall results in a low false negative rate. This helps us determine the quality of the output. The results are generated by varying the decision threshold of the model. This gives us more insights as to where an optimal confidence threshold to run inference at. An ideal model would be as close as possible to the top right corner across all thresholds. In our model, as we move along the curve there is a noticeable tradeoff between precision and recall. This tradeoff is to be expected. The wire cutter class performs the worst in this metric having its precision decrease the fastest at about 0.6 recall while the other classes perform more as expected. Overall our model performs well for this metric.

C. Real-World Human-Robot Collaboration

Fig. 7 shows us the real-world experimentation and process of the proposed approach. In each part, the top left shows us the YOLOv5 inference, the bottom left shows us a visualization of the robot in RViz and the right side shows us an external view of the workspace. Fig. 7 (1) shows the human collaborator prompts the robot for a screwdriver. In Fig. 7 (2), the robot has moved after scanning one side of the workspace to center itself with the screwdriver that was located on that initial side. In Fig. 7 (3), the robot then moves to grasp the tool and finally, in Fig. 7 (4) the robot moves to a designated handoff position to drop the tool for the human collaborator. Thus, showing the complete cycle of one prompt for a tool and this process would repeat as many times as needed. The full demonstration video shows another crucial aspect of this process where tool positions were switched around and the robot grasped the tools at different locations: https://youtu.be/ucAgSIK6crA. This is important for a real-world demonstration because tools may not stay in their original positions.

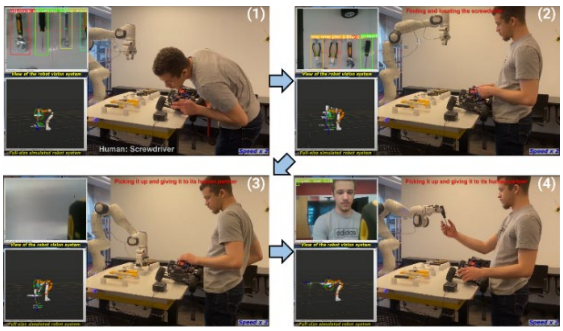

Fig. 7. Real-world human-robot collaboration shows the robot picks up one of the tools.

VI. Conclusions and Future Work

In this study, we have created an improved approach for the handover task in human-robot collaboration contexts. We proved that our framework of learning, finding, and giving combined with our specific approach resulted in a safe, effective, and repeatable way to conduct human-robot collaborative tasks. The major source of loss and error in our training is from false negatives. Our model could improve across all metrics. To do this as well as to scale our approach to a larger, more complex, manufacturing environment, we will improve our dataset by increasing images per class and instances per class. The variety of objects that we use would also need an increase in variability concerning certain aspects of luster, contamination, deterioration, and blemishes. Greater variety could also improve our dataset by capturing more images representative of real-world environments, different aspects of those environments such as lighting or different angles can help our model make more correct recognition. Moreover, adding background pictures or pictures with no

classifications in them is another recommended tactic to increase model performance. In the future, we can use these enhancements along with newer versions of YOLO or other object detection models to obtain better model performance. We can also create a larger experiment to test and improve our approach with other human participants.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation under Grant CMMI-2138351 and in part by the National Science Foundation under Grant CNS-2104742.

REFERENCES

[1] K. Zhang, F. Niroui, M. Ficocelli, and G. Nejat, "Robot Navigation of Environments with Unknown Rough Terrain Using Deep Reinforcement Learning," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1-7.

[2] J. Khurshid and B.-r. Hong, "Military robots - a glimpse from today and tomorrow," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, 6-9 Dec. 2004 2004, vol. 1, pp. 771-777.

[3] T. T. Nguyen, J. Parron, O. Obidat, A. R. Tuininga, and W. Wang, "Ready or Not? A Robot-Assisted Crop Harvest Solution in Smart Agriculture Contexts," in *2023 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2023, pp. 373-378.

[4] K. A. Kas and G. K. Johnson, "Using unmanned aerial vehicles and robotics in hazardous locations safely," *Process Safety Progress,* vol. 39, no. 1, p. e12066, 2020.

[5] S. Kim, "Working With Robots: Human Resource Development Considerations in Human–Robot Interaction," *Human Resource Development Review,* vol. 21, no. 1, pp. 48-74, 2022.

[6] F. Semeraro *et al*, "Human–robot collaboration and machine learning: A systematic review of recent research," *Robotics and Computer-Integrated Manufacturing,* vol. 79, p. 102432, 2023.

[7] H. Diamantopoulos and W. Wang, "Accommodating and Assisting Human Partners in Human-Robot Collaborative Tasks through Emotion Understanding," in *2021 International Conference on Mechanical and Aerospace Engineering (ICMAE)*, 2021, pp. 523-528.

[8] L. Scalera, A. Giusti, R. Vidoni, and A. Gasparetto, "Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones," *The International Journal of Advanced Manufacturing Technology,* vol. 121, no. 9, pp. 6783-6798.

[9] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating Human–Robot Collaborative Tasks by Teaching-Learning-Collaboration From Human Demonstrations," *IEEE Transactions on Automation Science and Engineering,* vol. 16, no. 2, pp. 640-653, 2018.

[10] W. Wang, R. Li, Y. Chen, Y. Sun, and Y. Jia, "Predicting Human Intentions in Human-Robot Hand-Over Tasks Through Multimodal Learning," *IEEE Transactions on Automation Science and Engineering,* vol. 19, no. 3, pp. 2339-2353, 2022.

[11] K. Strabala *et al.*, "Toward seamless human-robot handovers," *J. Hum.-Robot Interact.,* vol. 2, no. 1, pp. 112–132, 2013.

[12] S. Borgsen *et al*, "Improving Human-Robot Handover Research by Mixed Reality Techniques," in *IEEE/ACM International Conference on Human-Robot Interaction*, 2018.

[13] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, "Human-robot interaction in handing-over tasks," *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication,* pp. 107-112, 2008.

[14] W. Wang, R. Li, Y. Chen, and Y. Jia, "Human Intention Prediction in Human-Robot Collaborative Tasks," presented at the Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, Chicago, IL, USA, 2018.

[15] A. Castro, F. Silva, and V. Santos, "Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics," *Sensors,* vol. 21, no. 12, p. 4113, 2021.

[16] W. P. Chan, C. A. C. Parker, H. F. M. V. d. Loos, and E. A. Croft, "Grip forces and load forces in handovers: implications for designing human-robot handover controllers," the seventh annual ACM/IEEE international conference on Human-Robot Interaction, USA, 2012.

[17] C. Wang, C. Pérez-D'Arpino, D. Xu, L. Fei-Fei, K. Liu, and S. Savarese, "Co-GAIL: Learning Diverse Strategies for Human-Robot Collaboration," presented at the Proceedings of the 5th Conference on Robot Learning, Proceedings of Machine Learning Research, 2022.

[18] S. M. M. Rahman, Y. Wang, I. D. Walker, L. Mears, and R. Pak, "Trust-based compliant robot-human handovers of payloads in collaborative assembly in flexible manufacturing," *2016 IEEE International Conference on Automation Science and Engineering (CASE),* pp. 355-360, 2016.

[19] B. Sadrfaridpour *et. al*, "Modeling and control of trust in human-robot collaborative manufacturing," in *Robust Intelligence and Trust in Autonomous Systems*: Springer, 2016, pp. 115-141.

[20] A. Gómez Eguíluz, I. Rañó, S. A. Coleman, and T. M. McGinnity, "Reliable robotic handovers through tactile sensing," *Autonomous Robots,* vol. 43, no. 7, pp. 1623-1637, 2019.

[21] P. Rayamane, F. Munguia-Galeano, S. A. Tafrishi, and Z. Ji, "Towards Smooth Human-Robot Handover with a Vision-Based Tactile Sensor," presented at the Towards Autonomous Robotic Systems: 24th Annual Conference, TAROS 2023, Cambridge, UK, September 13–15, 2023.

[22] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 25-30 Sept. 2011 2011, pp. 1986-1993.

[23] S. Kumar *et. al*, "Survey of human–robot collaboration in industrial settings: Awareness, intelligence, and compliance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 51, no. 1, pp. 280-297, 2020.

[24] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics,* vol. 55, pp. 248-266, 2018.

[25] L. Peternel *et. al*, "Towards ergonomic control of human-robot co-manipulation and handover," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 55-60.

[26] S. El Zaatari *et. al*, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems,* pp. 162-180, 2019.

[27] M. Melchiorre, L. S. Scimmi, S. Mauro, and S. P. Pastorelli, "Vision-based control architecture for human–robot hand-over applications," *Asian Journal of Control,* vol. 23, no. 1, pp. 105-117, 2021.

[28] P. Rosenberger *et al.*, "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robotics and Automation Letters,* vol. 6, no. 1, pp. 17-23, 2020.

[29] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5," *Remote Sensing,* vol. 13, no. 9, p. 1619, 2021.

[30] S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, "Advanced active learning strategies for object detection," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020: IEEE, pp. 871-876.

[31] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," *IEEE International Conference on Robotics and Automation*, 2000, vol. 3: IEEE, pp. 2741-2746.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.

[33] J. Terven and D.-M. Cordova-Esparza, *A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond*. 2023.

[34] S. Chitta, I. Sucan, and S. Cousins, "Moveit!," *IEEE Robotics & Automation Magazine,* vol. 19, no. 1, pp. 18-19, 2012.

[35] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," *The International Journal of Robotics Research,* vol. 18, no. 3, pp. 286-298.

[36] S. Bier, R. Li, and W. Wang, "A Full-Dimensional Robot Teleoperation Platform," in *2020 IEEE International Conference on Mechanical and Aerospace Engineering*, 2020: IEEE, pp. 186-191.

[37] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2: Kobe, Japan, pp. 1-6.