Enhanced Human-Robot Collaboration through AI Tools and Collision Avoidance Control

Enhanced Human-Robot Collaboration through AI Tools and Collision Avoidance Control

Matteo Forlini
DIISM
Università Politecnica delle Marche
Ancona, Italy
m.forlini@pm.univpm.it

Federico Neri
DIISM
Università Politecnica delle Marche
Ancona, Italy
federico.neri@pm.univpm.it

Luca Carbonari
DIISM
Università Politecnica delle Marche
Ancona, Italy
luca.carbonari@staff.univpm.it

Massimo Callegari
DIISM
Università Politecnica delle Marche
Ancona, Italy
m.callegari@staff.univpm.it

Giacomo Palmieri
DIISM
Università Politecnica delle Marche
Ancona, Italy
g.palmieri@staff.univpm.it

***Abstract*—Due to the spread of collaborative robotics, human-robot collaboration is becoming more and more common in industrial environments. However, there is a need to make this interaction increasingly usable for the operator and less tiring by making the robot adapt to the operator's needs. In this paper, a framework based on Deep Learning techniques is presented that enables the operator to perform an manufacturing task assisted by a collaborative robot in a safer, less tiring and more flexible way. Three RGB-D cameras are used to capture information about the environment in which the human is working and about the human position. The framework has three main components: gesture recognition, robotic grasping and collision avoidance. Specifically, a Convolutional Neural Network has been implemented for gesture classification. Through gestures, the operator tells the robot what tool is needed for that subtask and the robot provides it. At the end of the machining process, through an automatic grasping pose detection, the robot is able to pick up the tool on its own at the position where the operator left it. During the interaction with the robot and the sharing of the workspace, the safety of the operator is ensured by avoiding collision with the robot. The safety distance between human and robot is always respected. Results of testing the framework on a real user-case using the Universal Robot 5e robot are presented.**

***Index Terms*—Robotics Collision Avoidance, Robotic Grasping, Gesture Recognition, Deep Learning, Human Robot Collaboration**

I. Introduction

Collaborative robotics represents a transformative approach to robotics aligned with the principles of Industry 4.0, facilitating symbiotic interactions between humans and robots [1]. Unlike traditional industrial robotics, collaborative robots operate in shared workspaces alongside humans, enabling tasks to be performed concurrently or in cooperation [2]. Essential to this paradigm are sensor systems integrated into robotic manipulators, enabling real-time detection of external contacts and averting potential collisions with humans [3]. Distinctive features of collaborative robots include rounded shapes, low payloads, and reduced speeds compared to industrial counterparts, ensuring intrinsic operator safety [4]. This innovation is rapidly spreading in various industries, enabling collaborative environments where humans and robots work together.

Operator well-being is crucial to achieve higher productivity, greater flexibility in performing tasks, and less material waste [5], so for this reason, is also important the robot perception of the external environment through exteroperceptive sensors. The robot thanks to them is able to perceive information from outside and through artificial intelligence techniques process it and use it to make interaction with the robot more usable. In this way the robot becomes an active intelligent machine capable of adapting to the operator's work rhythm and needs. The concept of human-machine interaction is reversed in this way, where it is no longer the machine that imposes the working conditions, having rigid automation, forcing the human to adapt to it, but it is the human that imposes working conditions and the machine by perceiving them with external sensors and processing them thanks to artificial intelligence software is able to adapt to it. The outcome is an enhancement of both physical and mental well-being for operators [6]. An illustrative application of collaborative robotics lies in assembly lines, where humans undertake tasks requiring flexibility and creativity, while repetitive and labor-intensive activities are delegated to robots [7]. In such human-robot interactions, considerations of ergonomic design and worker safety are mandatory. While current standards such as ISO/TS 15066 and ISO 10218-1 mandate robots to halt upon collision detection [3], [8], a more optimal solution would involve robots perceiving obstacles and autonomously avoid them. This approach ensures task completion without interrupting production, thereby enhancing both efficiency and operator safety. During a human-robot collaboration, an obstacle to avoid in order to make the collaboration more efficient, is the human body itself. To do this, the first step is to perceive the position of the human body relative to the robot. It is important

to know where each body joint is located and then implement an avoidance strategy aimed at maintaining a safe distance between the robotic arm and the human [9]. Two categories of exteroceptive sensors can be used to get in real time the position of the human body: wearable and non-wearable sensors [10]. Among unwearable sensors depth camera sensors (RGB-D) are the most popular, such as Microsot Kinect, Intel RealSense, Asus Xtion, etc.

To detect correctly the human body it is essential to use multiple cameras to prevent occlusion problems. Usually several cameras have to be set in the space in order to cover the principle viewpoints; then, a data fusion algorithm is needed [11], [12].

In [13] is used a kinect camera to detect a human hand and then implement an obstacle avoidance strategy based on kinetostatic receptive field.

In [14] is presented an algorithm based on controller barrier function around each link of the manipulator to avoid contact with the operator. It was used two RGB-D cameras to know the position of the human.

Flacco *et al.* [15] developed a method to rapidly merge data from multiple RGB-D cameras and calculate the distance between two points of interest (robot and human link). Then they implemented an avoidance algorithm based on repulsive vectors.

Nowadays, Machine Learning (ML) techniques [16], [17] and foundation models [18]–[20] are spreading in work environments where human-robot interaction is taken into account. In order to detect what is surrounding the robot, images are acquired and classified with Deep Learning tools or relying on the latest Visual Large Model systems [21], [22], which can understand both text and images and thus are widely used for images captioning techniques [23], visual question answer [24], [25] and optical character recognition. An important aspect of enabling a more fruitful interaction between human and robot concerns communication between the two actors. The human must be able to communicate to the robot what it needs at that point in the processing phase in a simple and effective way. To do this, input from the human is provided in several ways: through hand gestures [26], [27], voice commands [28], [29], facial expression [30]. In all these ways, machine learning or deep learning techniques are used to process the input data and extract useful information [31]. The grasping of objects by the robot in an autonomous and unsupervised manner is an important part of research on which many efforts are being focused today. In fact, this allows the operator to, for example, leave objects in a non-predefined position and then thanks to an RGB-D camera and Deep Learning algorithms the best grasp position will be identified based on the type of object and its location [32]. Several datasets are available for such training purpose [33].

In this paper, an application of collaborative robotics is presented in which there is an interaction between human and robot, making this collaboration more fruitful and less stressful for human using the above principles i.e. human obstacle avoidance, human gesture recognition and automatic object grasping. At the base of all three subfields there are artificial intelligence algorithms that receive information from one or more RGB-D cameras, process that information and provide output to the robot, which is then able to adapt to human movement and willingness. The purpose of the application is to simulate a component manufacturing process assisted by a collaborative robot. Specifically, the robot at the request of the operator will have to bring the required tool, communicated via gesture, to him at the required location, avoiding collisions with the operator along its trajectory. When the task is finished, the operator can leave the tool wherever on the work table not caring about its position, and the robot will pick it up automatically.

Specifically regarding collision avoidance, the authors developed a software framework in Python based on a machine learning tools for human skeleton detection (Mediapipe). Mediapipe is widely used in the field of human motion detection and its reliability is solidly proven [34], [35]. The framework allows an easy but robust implementation and integration of different RGB-D cameras of any brand and also an efficient fusion of the data provided by the cameras. Obstacles represented by the human are sent to an avoidance algorithm developed by the authors [36], [37], and a safe distance between the human and the entire kinematic chain of the robot is then ensured during the interaction, preventing contact and thus stopping the system. The robot is able to realtime replan the trajectory of the end-effector in order to avoid the obstacle but completing the planned task.

Gesture recognition was implemented through gesture classification done by Convulutional Neural Network. A dataset of gestures was acquired from different subjects. A neural network developed by the authors has been trained by performing hyperparameter optimization [38]. Each gesture corresponds to a tool for the robot to bring to the operator. By performing the gesture, it is thus possible to choose which tool is desired at that moment and in which position. In fact, through Mediapipe HandPose [39] when the gesture is made the landmarks of the hand are also identified and it is therefore possible to know the position of the hand with respect to the robot, which will bring the tool to that exact position.

For automatic grasping, a grasping detection based on Generative Residual Convolutional Neural Network has been implemented [40]. The X, Y, Z grasping pose with respect to the robot frame and the angle θ of gripper orientation with respect to the vertical axis are predicted.

The remainder of the manuscript is organized as follows: Section II presents the subparts of the framework as well as robotic grasping, gesture recognition and collision avoidance; experimental tests and results are described in Section III; a final discussion and some insights to future developments are given in Section IV.

II. Methods

Figure 1 illustrates the concept of the Human-Robot Collaboration Framework, showing two agents, a human and a robot, collaborating in machining or assembling a part.

The workspace is framed by three Intel Realsense D455 cameras, facilitating human-robot collaboration. Two cameras are dedicated to skeleton detection, providing real-time human position data to the avoidance algorithm, while the third captures RGB images for gesture classification and RGB-D images for object grasping pose identification. Upon the arrival of the workpiece, the operator selects a tool from the two options held by the robot's dual gripper by performing a gesture, either open palm or closed fist, which is captured by a dedicated camera positioned perpendicular to the work surface. A Convolutional Neural Network (CNN) classifies the gesture, and via TCP-IP socket communication, instructs the robot on which tool to provide to the operator. Simultaneously, the gesture's location relative to the robot is determined using RGB and Depth frames, ensuring precise tool delivery to the operator's hand. Subsequently, the operator starts machining with the provided tool. Upon completion, the tool is released onto the workbench, automatically identified for grasping pose by the robot, and returned to the gripper. Throughout these actions, the avoidance algorithm operates, pausing only when the robot approaches the final tool release position near the operator's hand to allow for a reduced safe distance between the robot's hand and the tool.

A. Robotic Grasping

To grasp the tool at the end of the process a Generative Residual Convolutional Neural Network (GRCNN) developed by [40] and trained on the Cornell grasp and Jacquard grasping dataset is used. The input of the network is an RGB-D 640×480 image and gripper amplitude. This way, only grasping poses realizable by the gripper that is available, are provided. The image is then processed by an inference module before being passed to GRCNN. The output is the identification of the grasping rectangle that simulates the position of the gripper. The grasping pose has 4 degrees of freedom which are X, Y, Z and the angle θ which is the orientation of the gripper with respect to the vertical axis (the axis of the gripper is perpendicular to the working plane). It is given which is the central u, v pixel of the rectangle and the angle of orientation θ ranging from $-\pi/2$ to $+\pi/2$. Knowing the pixel and reading the distance corresponding to that pixel, it is possible first through the intrinsic parameters of the camera to know the grasping position X, Y, Z in mm with respect to the camera and then through the calibration of the camera, which provides the extrinsic parameters, to transform that pose into the robot's reference system. The calibration procedure that is common for both tool grasping and skeleton detection is illustrated in Subsection II-B. The pose X, Y, Z, θ with respect to the robot frame is communicated to the robot via TCP-IP, which picks up the object, visible in Figure 2. The grasping rectangle is shown in the Figure 3. As can be seen from Figure 2 the system has also been tested when there are more items: it is able to identify which is the most convenient first object to take if there are multiple objects on the workbench.

B. Camera Calibration

The calibration process for the RGB sensor involved several steps. Initially, a standard calibration was conducted using the Matlab toolbox. Images of a checkerboard were captured within the camera's field of view, with one image featuring the checkerboard at a known location relative to the robot base. This step allowed for an estimation of the relative pose between the robot and the camera. Following this, a hand-eye calibration process was carried out. A custom tool with a spherical tip was mounted onto the robot's end-effector and moved to 12 distinct positions within the camera's frame. The 3D coordinates of the spherical tip were known from the manipulator's kinematic model. Using the estimated intrinsic and extrinsic parameters, the 3D metric coordinates of the tip were projected onto the RGB plane, generating pixel coordinates. These coordinates were compared with the point corresponding to the centroid of the tip in the image. The cumulative error between the projected and framed centroids across all 12 tip positions served as the objective function for a minimization procedure, employing the fminsearch algorithm in Matlab, to optimize the extrinsic parameters of the RGB sensor. The quality of calibration was assessed in terms of reprojection error, which remained below 0.8 pixels for all cameras. Furthermore, in terms of real-world X, Y, Z coordinates and given the pixel coordinates u, v of a point, along with its distance from the camera as determined by the depth sensor (averaging 890 mm), the 3D position of that point could be estimated with an error lower than 3 mm..

C. Gesture Recognition

Gesture recognition involves a binary classification between two types of gestures based on CNN. A customized dataset was acquired on the two gesture types, which are palm open and fist closed. Each gesture corresponds to one of the two grippers mounted on the robot's end effector (visible in Figure 2. The dataset was acquired with the actual setup then used in the final experimental part. Three different subjects were involved in the acquisition; 380 images were captured for each of them, 190 for each class. The total number of images is 1140. For training, 80% of them were adopted. The captured images have a size of 100×100 pixels in black and white (1 channel). The black-and-white images, subsequently have been binarized. The CNN has been developed using a sequential model consisting of the following layers shown in the Table I. Several feature possibilities were tested by hyperparameters optimization using the Hyperband [38] algorithm.

The loss is computed as BinaryCrossentropy and the Adam optimizer is used with different learning rates tested always as hyperparameters assuming values of $10^{-2}, 10^{-3}, 10^{-4}$

The other hyperparameters to be tested assume the different possibilities given in the Tabel II.

Using the Hyperband algorithm with a maximum number of epochs equal to 100 and factor 3, the best validation accuracy of 96.7% is obtained with the hyperpameters given in Table III.

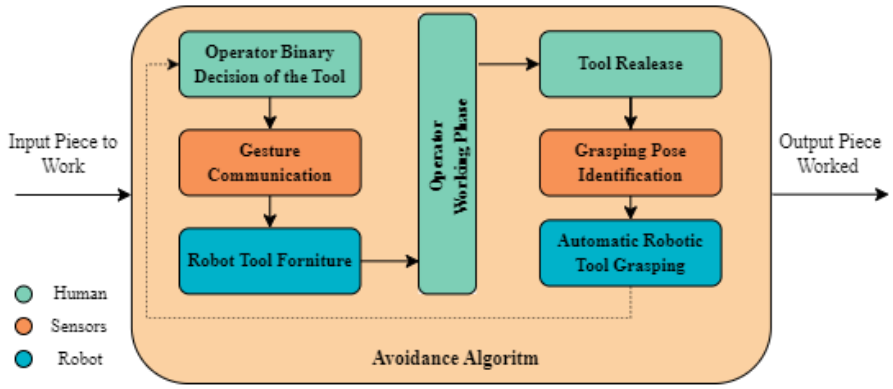

Fig. 1: Concept of the Human-Robot collaboration Framework

TABLE I: Convolutional Neural Network Architecture

Layer	Number Features	Kernel Size	Pool Size	Unit	Activation	Kernel Regularization
Conv2D	*hyperparam_1*	5×5	-	-	Relu	*hyperparam_5*
MaxPooling2D	-	-	2×2	-	-	-
Conv2D	*hyperparam_2*	3×3	-	-	Relu	*hyperparam_5*
MaxPooling2D	-	-	2×2	-	-	-
Conv2D	*hyperparam_3*	3×3	-	-	Relu	*hyperparam_5*
MaxPooling2D	-	-	2×2	-	-	-
Flatten	-	-	-	-	-	-
Dense	-	-	-	*hyperparam_4*	Relu	-
Dense	-	-	-	1	Sigmoid	-

TABLE II: Hyperparameters Optimization

Hyperparameter	Options
hyperparam_1...3	32, 64, 96, 128
hyperparam_4	32, 64
hyperparam_5	True, False

TABLE III: Results of Hyperparameters Optimization

Hyperparameter	Value
hyperparam_1	128
hyperparam_2	96
hyperparam_3	64
hyperparam_4	64
hyperparam_5	True
learning_rate	10^{-2}

D. Skeleton Detection and Avoidance Algorithm

Utilizing two Intel Realsense D455 RGBD cameras, human identification is achieved, capturing the entire scene from various perspectives and mitigating occlusion issues effectively. The data stream from both the depth sensor and the RGB image sensor is configured at a resolution of 424x240 pixels. MediaPipe Pose serves as a machine learning solution for precise body pose tracking, extracting 33 3D landmarks across the entire body from RGB video frames [41]. For each landmark, denoted by ID_i, its x, y pixel coordinates on the image are provided by MediaPipe. The corresponding z coordinate is obtained from the depth frame for the respective pixel. Using x, y pixel coordinates and z value is possible to reconstruct the 3D spatial position of the human joint respect to the robot thanks to the camera calibration. Notably, MediaPipe assigns a visibility value, $ID_{visibility}$, to each landmark, indicating the probability of occlusion. Only landmarks with a $ID_{visibility}$ exceeding 0.95 are considered, discarding the rest. The camera with the maximum $ID_{visibility}$ value for each landmark is selected. Moreover, the Mediapipe parameters "min detection confidence" and "min tracking confidence" are both set to a high value of 0.90, bolstering the robustness of the machine learning algorithm. This ensures that the system distinguishes human bodies from robots during detection and selects the camera with minimal occlusions effectively. Skeleton detection is performed by a standard PC, and the positions of human joints relative to the robot reference frame are transmitted via TCP/IP at 18 Hz to a secondary PC. This PC executes the obstacle avoidance algorithm and dispatches velocity reference signals to the robot controller using UR RTDE interface [37].The overall frequency of the avoidance controll is about 200 Hz. For further details regarding the avoidance algorithm, refer to [42]. During testing, the motion tracking system provides coordinates for all visible joints, which are then forwarded to the obstacle avoidance algorithm. For simplicity, the algorithm considers only the landmarks of the right and left wrists. This decision stems from the likelihood of these body parts coming into contact with the robot during a human-robot

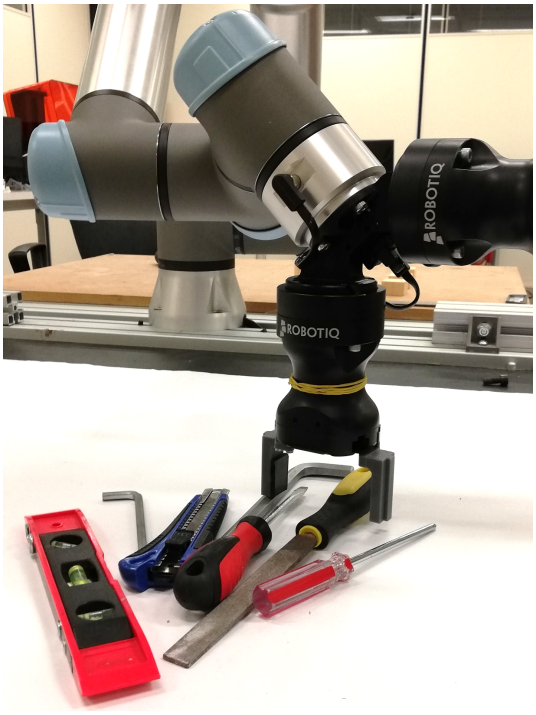

Fig. 2: Robot in position of grasping

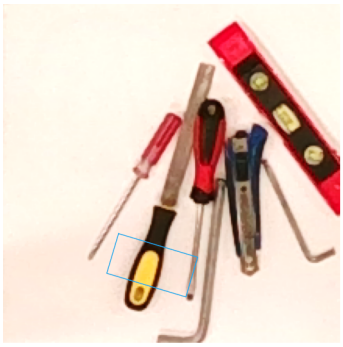

Fig. 3: Grasping pose identification

collaboration scenario in an assembly task, given the test setup. Key parameters for the obstacle avoidance algorithm are set as follows: $r = 0.20, \mathrm{m}$, $v_{rep} = 0.40, \mathrm{m/s}$, $\dot{\theta}max = \pi/2, \mathrm{rad/s}$. Here, r represents the safety distance from the robot, $vrep$ denotes the repulsive velocity applied to the robot to avoid obstacles, and $\dot{\theta}_{max}$ signifies the maximum joint speed permissible for the robot. These parameter values were chosen through trial and error to ensure collision avoidance during typical human movement speeds while preventing the robot from reacting excessively fast, which could induce cognitive stress on the operator.

III. Experimental Tests and Results

Experimental test and results will be presented in the final extended version of the paper. The main concept is to present several tests with different subjects cooperating with the robot and using the framework described above involving all the different aspects from grasping, gesture recognition, avoidance with skeleton. For the avoidance part will be shown a graph illustrating that the distance between robot body will remain upper than the safety distance. Additionally, the outcomes of the grasping and gesture recognition tests will be summarized in a table, which will catalog instances of success and failure, accompanied by explanations for each result.

(a) Open Palm. (b) Close Fist.

Fig. 4: Binarized gestures images.

IV. Conclusions

A discussion of the results, highlighting both critical limiting and strengths aspects will be written. Future developments will be summarised.

References

[1] H. Kagermann, W. Wahlster, J. Helbig, *et al.*, "Recommendations for implementing the strategic initiative industrie 4.0," *Final report of the Industrie*, vol. 4, no. 0, p. 82, 2013.

[2] H. Liu, T. Fang, T. Zhou, and L. Wang, "Towards robust human-robot collaborative manufacturing: Multimodal fusion," *IEEE Access*, vol. 6, pp. 74762–74771, 2018.

[3] ISO organization, *ISO/TS 15066:2016 Robots and robotic devices — Collaborative robots*, 2016.

[4] "Collaborative industrial robot definition and estimates supply," 2019. International Federation of Robotics Secretariat Blog.

[5] L. Probst, L. Frideres, B. Pedersen, and C. Caputi, "Service innovation for smart industry: human–robot collaboration," *European Commission, Luxembourg*, 2015.

[6] L. Gualtieri, I. Palomba, F. A. Merati, E. Rauch, and R. Vidoni, "Design of human-centered collaborative assembly workstations for the improvement of operators' physical ergonomics and production efficiency: A case study," *Sustainability (Switzerland)*, vol. 12, no. 9, 2020.

[7] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP annals*, vol. 58, no. 2, pp. 628–646, 2009.

[8] ISO organization, *ISO 10218-1:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots*, 2011.

[9] P. A. Lasota, T. Fong, J. A. Shah, *et al.*, "A survey of methods for safe human-robot interaction," *Foundations and Trends® in Robotics*, vol. 5, no. 4, pp. 261–349, 2017.

[10] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for collaborative robots: Fundamentals, challenges, and opportunities," *Frontiers in Neurorobotics*, p. 113, 2021.

[11] C. Lenz, M. Grimm, T. Röder, and A. Knoll, "Fusing multiple kinects to survey shared human-robot-workspaces," 2012.

[12] P. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl, and R. Simmons, "Sensor fusion for human safety in industrial workcells," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3612–3619, IEEE, 2012.

[13] M. P. Polverini, A. M. Zanchettin, and P. Rocco, "A computationally efficient safety assessment for collaborative robotics applications," *Robotics and Computer-Integrated Manufacturing*, vol. 46, pp. 25–37, 2017.

[14] F. Ferraguti, C. T. Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robotics and Autonomous Systems*, vol. 124, p. 103388, 2020.

[15] F. Flacco and A. De Luca, "Real-time computation of distance to dynamic obstacles with multiple depth sensors," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 56–63, 2016.

[16] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran, "A survey of robot learning strategies for human-robot collaboration in industrial settings," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102231, 2022.

[17] M. Zamora, E. Caldwell, J. Garcia-Rodriguez, J. Azorin-Lopez, and M. Cazorla, "Machine learning improves human-robot interaction in productive environments: A review," in *International Work-Conference on Artificial Neural Networks*, pp. 283–293, Springer, 2017.

[18] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[19] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

[20] J.-W. Kim, J.-Y. Choi, E.-J. Ha, and J.-H. Choi, "Human pose estimation using mediapipe pose and optimization method based on a humanoid model," *Applied Sciences*, vol. 13, no. 4, p. 2700, 2023.

[21] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, *et al.*, "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.

[22] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[23] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson, "Nocaps: Novel object captioning at scale," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8948–8957, 2019.

[24] X. Hu, Z. Gan, J. Wang, Z. Yang, Z. Liu, Y. Lu, and L. Wang, "Scaling up vision-language pre-training for image captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17980–17989, 2022.

[25] R. Zellers, X. Lu, J. Hessel, Y. Yu, J. S. Park, J. Cao, A. Farhadi, and Y. Choi, "Merlot: Multimodal neural script knowledge models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23634–23651, 2021.

[26] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[27] E. Coupeté, F. Moutarde, and S. Manitsaris, "Gesture recognition using a depth camera for human robot collaboration on assembly line," *Procedia Manufacturing*, vol. 3, pp. 518–525, 2015.

[28] I. Maurtua, I. Fernandez, J. Kildal, L. Susperregi, A. Tellaeche, and A. Ibarguren, "Enhancing safe human-robot collaboration through natural multimodal communication," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, IEEE, 2016.

[29] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, "An extensible architecture for robust multimodal human-robot communication," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2208–2213, IEEE, 2013.

[30] L. Chen, M. Zhou, W. Su, M. Wu, J. She, and K. Hirota, "Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction," *Information Sciences*, vol. 428, pp. 49–61, 2018.

[31] L. Wang, R. Gao, J. Váncza, J. Krüger, X. V. Wang, S. Makris, and G. Chryssolouris, "Symbiotic human-robot collaborative assembly," *CIRP annals*, vol. 68, no. 2, pp. 701–726, 2019.

[32] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, *et al.*, "Deep learning approaches to grasp synthesis: A review," *IEEE Transactions on Robotics*, 2023.

[33] Y. Huang, M. Bianchi, M. Liarokapis, and Y. Sun, "Recent data sets on object manipulation: A survey," *Big data*, vol. 4, no. 4, pp. 197–216, 2016.

[34] A. K. Singh, V. A. Kumbhare, and K. Arthi, "Real-time human pose detection and recognition using mediapipe," in *International Conference on Soft Computing and Signal Processing*, pp. 145–154, Springer, 2021.

[35] C. A. Q. Bugarin, J. M. M. Lopez, S. G. M. Pineda, M. F. C. Sambrano, and P. J. M. Loresco, "Machine vision-based fall detection system using mediapipe pose with iot monitoring and alarm," in *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 269–274, IEEE, 2022.

[36] M. Forlini, F. Neri, C. Scoccia, L. Carbonari, and G. Palmieri, "Collision avoidance in collaborative robotics based on real-time skeleton tracking," in *International Conference on Robotics in Alpe-Adria Danube Region*, pp. 81–88, Springer, 2023.

[37] F. Neri, M. Forlini, C. Scoccia, G. Palmieri, and M. Callegari, "Experimental evaluation of collision avoidance techniques for collaborative robots," *Applied Sciences*, vol. 13, no. 5, p. 2944, 2023.

[38] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.

[39] "Mediapipe, hand detection landmark." https://developers.google.com/mediapipe/solutions/vision/hand_landmarker, 2024.

[40] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9626–9633, IEEE, 2020.

[41] "Mediapipe, pose detection." https://google.github.io/mediapipe/solutions/pose.html, 2023.

[42] G. Chiriatti, G. Palmieri, C. Scoccia, M. C. Palpacelli, and M. Callegari, "Adaptive obstacle avoidance for a class of collaborative robots," *Machines*, vol. 9, no. 6, p. 113, 2021.