

Development and Implementation of an AI-Embedded and ROS-Compatible Smart Glove System in Human-Robot Interaction

Laury Rodriguez
Department of Computer
Science
Montclair State University
Montclair, USA
rodriguez116@montclair.edu

Zofia Przedworska
Department of Computer
Science
Montclair State University
Montclair, USA
przedworskaz1@montclair.edu

Omar Obidat
Department of Computer
Science
Montclair State University
Montclair, USA
obidato1@montclair.edu

Jesse Parron
Department of Computer
Science
Montclair State University
Montclair, USA
parronj1@montclair.edu

Weitian Wang*
Department of Computer
Science
Montclair State University
Montclair, USA
wangw@montclair.edu

Abstract—Robotics technology is being widely used for an array of tasks in today’s evolving markets. Human-robot collaboration is inevitable which leads to the need for safe, untroublesome, and easy-to-produce products. A smart glove has capabilities to collect data concerning its wearer’s movements by the use of sensors. Motivated by this, in this study, we develop an AI-embedded and ROS-compatible smart glove system to realize real-time human-robot interaction in collaborative tasks. To allow the robot to intelligently learn and predict new human intentions for human-robot interaction, we propose an Extreme Learning Machine (ELM)-based human gesture understanding approach using the data from a set of strip and force sensors embedded in the smart glove and effectively run it through ROS. Three typical baseline gestures are conjured for ELM training purposes and fed into the algorithm with an appended label and corresponding sensor data. The developed system and proposed approach are validated in real-world human-robot collaborative tasks with efficiency and success. This work can also serve as a catalyst for the implementation of many important robot-supported applications such as healthcare and daily assistance for senior groups. Future work of this study is also discussed.

Keywords—smart glove, artificial intelligence (AI), human-robot interaction, robot operating system (ROS), sensors, machine learning

I. INTRODUCTION

Human-robot interaction relies on the robots’ seamless interpretation of human actions. Robots that work with humans need to be equipped with the ability to dynamically make decisions when differentiated tasks are presented to them [1-3]. This can be encouraged by developing systems with the purpose of tracking and interpreting user input so that robots can be programmed with appropriate corresponding reactions [4-8]. A smart glove has capabilities to collect data concerning its wearer’s movements by the use of sensors [9, 10]. This data is then able to be processed and used to categorize the user input into a set of known gestures. The recognition of different hand gestures has the potential to be used in a variety of human-robot collaborative tasks, including medical procedures, product assembly, object hand-over, and kinaesthetic communication [11-15]. As more human hand movements are introduced, machine learning algorithms can be employed to recognize possible new gestures through the sensor data. A library of human hand movements has applications in equipping the robot with intention prediction competencies which it can employ in

real-time and without the need for the programming of its every action [16]. Such measures allow for the successful execution of human-robot interactions to which the robot can efficiently adapt.

Previous work has recorded efforts to enhance human-robot collaboration by studying human input in a given situation and shaping robots’ actions based on it. Cakmak *et. al.* investigated the role of nonverbal communication in facilitating human-robot interaction in an assembly line setting. This was observed to have an effect on the way robots perform tasks involving giving objects to humans [17]. Another study that focused on such interactions was conducted by Aleotti *et. al.*, who used Kinect body tracking technology to make a robot assess the most human-friendly way in which it could act [18]. The Kinect method detected when a human hand was holding an object, although the system was designed to track the placement of a handover object relative to the hand of its recipient, rather than independently evaluating the various motions made by the human hand. Yacoub *et. al.* used a Random Forest algorithm to have a robot’s actions mimic those of humans [19]. The system relied on Force/Torque sensor data to program the moves made by robots in co-manipulative settings. The study has applications limited to situations where humans and robots simultaneously work on a task. Cohen *et. al.* created an interface that employed intention prediction methods. This system relied on the close monitoring of human actions to foresee the user’s next choice [16]. As the system operates without assigning values to human movements, it requires a given amount of interaction with a user before starting to make decisions. Yu *et. al.* used the Lyapunov theory to aid in the filling in of unstable output from robots [20]. The robots subject to this experiment were equipped with torque sensors, and computers were used to approximate data such as the velocity of the robots’ movements. The study was conducted using a system compatible with Baxter Robot Operating System SDK (RSDK). Yahya *et. al.* shortened the average training time for several robots learning how to complete a task. The study relied on a modified version of Guided Policy Search algorithms to reduce issues caused by processing the data from the robots [21].

Different from the above research efforts, in this study, we develop an AI-embedded and ROS-compatible smart glove system to realize real-time human-robot interaction in collaborative tasks. To combat the gap in real-time and intelligent gesture recognition techniques for human-robot interaction, we propose an Extreme Learning Machine (ELM)-based human gesture

understanding approach using the data from a set of strip and force sensors embedded in the smart glove and effectively run it through ROS. To achieve this, three typical baseline gestures are conjured for ELM training purposes and fed into the algorithm with an appended label and corresponding sensor data. This is considered offline data as it is pre-processed and not fed in real-time. The ultimate goal is to have the trained ELM recognize in real-time gestures being performed by the user of the smart glove system and in turn interact with robots based on the results of the model. After the initial training, a ROS subscriber node embedded within the ELM would begin receiving online data from a ROS publisher script. The trained algorithm would be equipped enough to output its prediction of the gesture being motioned by the user. The predicted human intentions would be used in human-robot interaction via another ROS node. The developed system and proposed approach are validated in real-world human-robot collaborative tasks with efficiency and success. This work can also serve as a catalyst for the implementation of many important use cases. With the ROS embedded feature, this ELM technique is easily modifiable for use in various practical applications. The contributions of this work can be summarized as: (1) We develop an easy-to-use smart glove system, which is compatible with ROS and can be used for real-time human-robot interaction; (2) An AI-embedded human gesture understanding approach is proposed to allow the robot to intelligently learn and predict new human intentions for human-robot interaction.

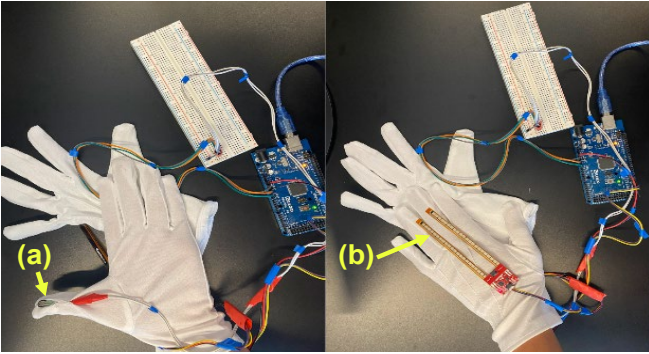


Fig. 1. The smart glove system.

II. SYSTEM DEVELOPMENT

A. System Overview

As shown in Fig. 1, the smart glove system consists of a force-sensitive resistor (FSR) [22] (Fig. 1(a)) and a set of Spark Fun flex sensors [23] (Fig. 1(b)) integrated into a glove. The sensors sit flush with the assistance of another glove overtop. The FSR is sewn over the thumb and the strip sensors on top of the hand over the pointer and middle fingers. Although the unit for the flex sensors includes a controller breakout, jumper cables were soldered directly into the board's pins for ease of use, as presented on the right side of Fig. 1. The force sensor used is an Interlink model with its two extended pins can be directly placed into a breadboard. Nonetheless, this would render it unusable in a glove system. As such, the FSRs were connected via female jumper cables to the Arduino microcontroller. The Arduino Mega2560 used has 54 digital input/output pins, 16 analog inputs, and 4 serial ports [24]. With this board, communication between the sensors is achieved,

allowing the smart glove system's sensors to output data. The system is easy to use and construct. After attaching all components and with the correct software, the user can begin to see in the terminal the output of data from the sensors.

B. Flex Sensor

On one of the sides of the flex sensor, there is a print done with a polymer ink that has conductive particles embedded in it. For example, when the sensor is straight, its resistance is decreased. On the other hand, when the sensor is bent the resistance will be increased. The sensing information when the sensors are bent is calculated via the resistance as:

$$V_{o-flex} = V_{cc} * (R / (R + R_{flex})) \quad (1)$$

where V_{o-flex} is the output voltage, V_{cc} is the input voltage, R is the resistance of the pull-down resistor, and R_{flex} is the resistance between the ink and the particles.

As shown in Fig. 2, the Qwiic Flex Sensor [23], which is a durable more permanent solution to the traditional and fragile flex sensors seen on the market, is used in our smart glove system development. The controller integrated with the sensor keeps the solder points isolated to minimize breakage although spaced pins are available for soldering and/or use of a breadboard. The controller is two flex sensors joined at the head via a board. The board gives access to four pins: a 3.3-volt power supply, GND, SDA, and an SCL pin. Communication to the controller is enacted via an onboard ADS1015 ADC to I2C chip [23]. Connecting the sensor to a microcontroller can be achieved either with the proprietary Qwiic cable into the controller's breakout or through solder points found at the head.



Fig. 2. The flex sensor used in our smart glove system [23].

C. Force Sensor

We used a Force Sensitive Resistor (FSR) on the thumb of our smart glove system to track when a user wearing the glove applied pressure to the sensor, as presented in Fig. 3. The sensor is constructed of two flexible, semiconducting silicon wafers conjoined by an adhesive. The part that detects the applied force, known as the active area, consists of an interlocking electrode pattern and is cushioned by the substrate layers. Upon the application of force onto the electrodes of the FSR's active area, the resistance level drops from its baseline state [22]. The acquired information when force is applied on the sensor is estimated via the resistance as:

$$V_{o-FSR} = V_{cc} * (R / (R + R_{FSR})) \quad (2)$$

where V_{o-FSR} is the output voltage, V_{cc} is the input voltage, R is the resistance of the pull-down resistor, and R_{FSR} is the resistance of the force sensitive resistor. In the process of data collection, the object placed on the active, sensing area of the sensor is a human thumb. Measures were taken in ensuring consistency in the sensor reading between each placement of the human user's thumb on the sensor. For example, the human

user's thumb evenly places at the center of the sensor each time and avoids the adhesive-bound borders of the active area.

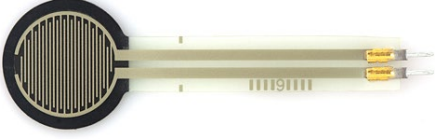


Fig. 3. The force sensor used in our smart glove system [22].

D. Software Architecture

Fig. 4 depicts the baseline software architecture designed for the smart glove system. The sensors simultaneously communicate with the Arduino board loaded with data collection code to output onto one line of data from the two sensors. Note, since the strip sensors are, in actuality, two conjoined sensors, its output is two pieces of data, one for each finger. From there, a ROS Publisher node listens for the conjoined data and the subscriber node written within the ELM receives said data. The trained ELM model accepts this data as its new test case and begins gesture recognition in real-time.

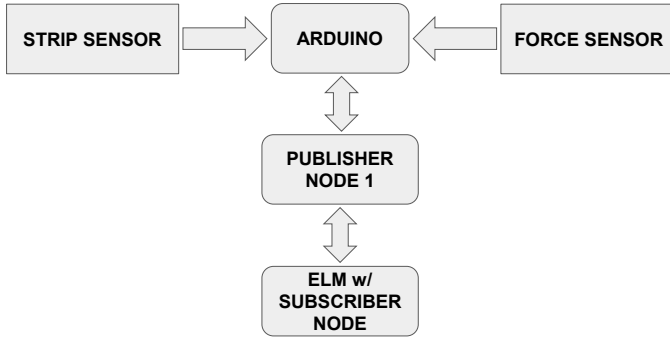


Fig. 4. Software architecture of the smart glove system.

Fig. 5 shows the extended software architecture for the smart glove system as applied to a collaborative robot. The architecture takes the original model and adds additional publisher and subscriber ROS nodes. In other words, after the trained ELM model categorizes the data subscribed from the microcontroller and sensors into gestures, it publishes its classifications and the additional subscriber node within an external python script acknowledges it. From here, the collaborative robot in conjunction with the control program performs the required action based on the gesture being performed in real-time.

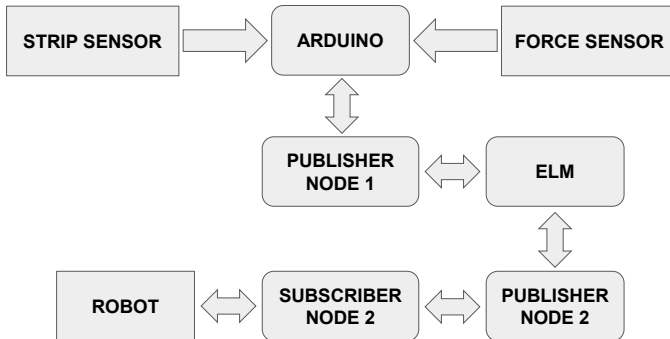


Fig. 5. Extended software architecture of the smart glove system.

E. Robot Operating System

The Robot Operating System suite was used in our study and was run on Ubuntu 20.04. ROS functions as a link between the raw sensor data and the controls that cause a robot to complete a set of given tasks [25-27]. ROS is a versatile body for prototype robotics software, which is compatible with several multiple languages and can be used with different hosts in a network rather than being restricted to the control of a single main server. ROS contains the algorithms necessary for the robot to function in the form of nodes which can freely exchange information with one another. Additional software distributed by ROS can be installed and added to the system in the form of individual packages. A standard ROS package is constructed from a directory that is accompanied by a listing of the package dependencies [25]. The broad applications of the ROS operating system were favorable in our study, as the smart glove system spans a variety of different configurations and real-world applications.

III. GESTURE RECOGNITION USING ELM

In machine learning, neural networks are a subset of learning that mimic the biological network system of the brain. Extreme Learning Machine (ELM) is an algorithm for single-hidden layer feedforward neural networks (SLFNs) with random hidden nodes [28-31]. This algorithm provides simpler learning methods that in turn lead to faster training times and good performance.

In ELM, the input layer is randomly assigned weights. In turn, the output weights are typically calculated using the least-squares (LS) solution. ELM operates much faster than its gradient-based competitors such as back propagation (BP) because of the lack of iteration in the learning process [30-32]. This is what makes ELM extremely fast and useful for many real-world applications, especially in the field of robotics.

The training methodology for ELM is straightforward. Given training set $N = (x_i, t_i)$, where $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$, activation function $g(x)$, and hidden node quantity N , the learning method via ELM is described as follows [31, 33]:

- (1) Creation of input weight w_i and bias b_i within $[-1, 1]$ range using activation functions;
- (2) Compute the hidden layer output matrix H ;
- (3) Compute the output weight β where $\beta = H^+T$, where H^+ is the generalized inverse of the output matrix H .

After the implementation of the learning method, the prediction result will be calculated from the maximum elements of $y_i = H\beta_k, k = 1, 2, \dots, C$. C denoting the number of classes.

For our application, the sigmoid activation function was chosen although any activation function is acceptable dependent on the task and data at hand. The model originally returned an accuracy score for both the training and testing stages. At its peak, the model reached a nearly 94% accuracy rate. At this stage, it was ready to replace its testing case with the real-time ROS node's data. Instead of returning success rates, the model will display the label of the gesture it believes. In other words, it will output the results of its prediction model. As such, gesture recognition is achieved.

IV. RESULTS AND ANALYSIS

A. Experimental Setup

The experimental platform was formed by our developed smart glove system and a workstation ThinkStation P520. The goal was to have the trained ELM algorithm recognize in real time the gestures a user of the smart glove system was exerting. For this experiment, three users demonstrated the three gestures (relaxed, fist, and pinch, as shown in Fig. 6) 50 times each and one at a time. The trained algorithm would output the gesture it believes is being displayed and the overseer would monitor activity from the model.

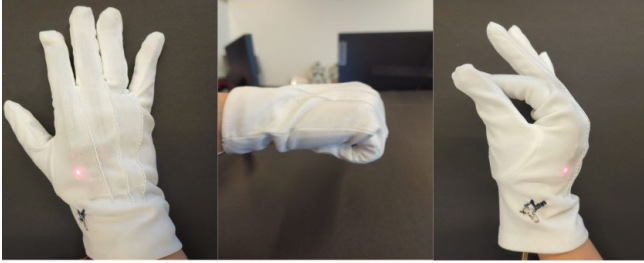


Fig. 6. Relax, fist, and pinch gestures.

There exists a large number of hand gestures humans conduct daily. When deciding on the gestures to be chosen for this experiment, it was imperative to keep in mind the objective of the project and its potential future uses. Ultimately, the three typical gestures chosen were a relaxed state, a fist, and a pinch as seen above in Fig. 6. Although future works can easily include more.

Gesture 0, the resting palm, consists of minimal to no bent of the strip sensors and no applied force to the FSR. Although strip sensors measure the bend of the finger, it's best practice to not try and spread the fingers too far either as it can slightly alter results in the data stream. Gesture 1 is a standard fist. For control purposes, the thumb with its attached FSR sensor rested outside of the fist as clearly seen in Fig. 7. The final gesture demonstrates a pinching motion modified so that one or both of the pointer and middle fingers are acceptable.

B. Gesture Recognition Results and Analysis

Table I shows the accuracy rates of our gesture recognition experiment for the right-hand glove. Three different users modeled the gestures and the ELM model's results were recorded for every 50 predictions per gesture. The machine was most confident with classifying gestures 1 and 2, fist and pinch, respectively. For user one, the ELM recognized accurately a fist gesture approximately 91.33% of the time. For the other users, it yielded 100% accuracy. Furthermore, gesture 2, the pinch, was the most accurate at 100% accuracy for all three users. The relaxed gesture, on the other hand, produced poorer results. At its best, the ELM accurately classified a relaxed state roughly 72.92% of the time. This is in contrast to a 52.62% accuracy rate for its worst case.

Some explanations for these varying results could lie within the definition of a neutral hand gesture. For this experiment, the participants were instructed to relax their hands and have minimal to no movement. As a matter of fact, for some this meant nearly straight fingers and for others a slight bend to the

hand was visible. Although visibly minuscule, this distinction could lead to varying data points from the strip sensors specifically that could lead to errors in the model. A possible solution could be to modify the gesture itself or train the model with more data from different users. This way, the model can be better equipped to recognize the relaxed hand gesture for different users.

Table I. Gesture recognition results.

	RELAXED - 0	FIST - 1	PINCH - 2
USER 1	69.17%	91.33%	100.00%
USER 2	52.62%	100.00%	100.00%
USER 3	72.92%	100.00%	100.00%

V. APPLICATION IN HUMAN-ROBOT INTERACTION

A. Task Design

We validated the developed smart glove system in a human-robot interactive context, which includes a collaborative robot, the smart glove, a target object, and a shared workspace. The robot used was the Franka Emika Panda, which is a robotic arm with 7-Degrees of Freedom (DoF) [27, 34]. It is a robot that is equipped with safety features that enable it to work closely with humans in a cooperative environment. During the experiment we tested 3 main cases:

- (1) Controlling the gripper of the collaborative robot through the smart glove;
- (2) Robot learning from the glove user to pick up an object;
- (3) Receiving an object delivered from the collaborative robot.

B. Results and Analysis

Fig. 7 presents the user who controls the gripper of the collaborative robot through our developed smart glove system. When the user's hand is relaxed, the smart glove recognizes this gesture state using the trained ELM model and tells the robot via ROS. As shown in the first picture in Fig. 7, the robot opens its gripper. In addition, once the pinch gesture is done, the robot closes its gripper based on the received gesture information from the smart glove. This kind of application would allow the user wearing the smart glove to control the picking and dropping of an object in human-robot collaborative tasks such as product co-assembly.

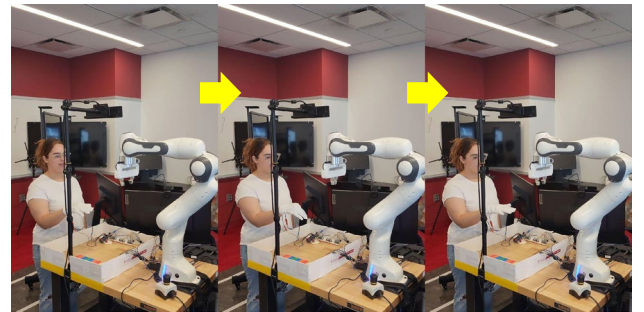


Fig. 7. Controlling the gripper of the collaborative robot through the smart glove.

Fig. 8 shows the results of the robot learning from the glove user to pick up an object. When the user's hand is pinching an object, the collaborative robot successfully moves to pick up a similar object from in front of it through the gesture recognition

results of the smart glove. This application would allow the user to control the collaborative robot and have it perform the same actions of the glove user in robot-assisted teleoperating tasks such as remote-healthcare and remote-surgeries.

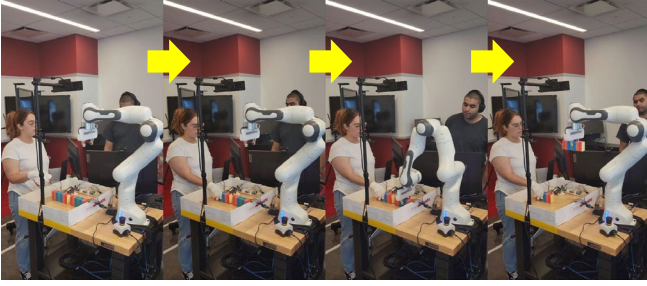


Fig. 8. Robot learning from the glove user to pick up an object.

In Fig. 9, when the user's hand is receiving an object delivered by the collaborative robot, the robot would release the object. Indicating that it receives the user's gesture information that the smart glove has secured the object and it is safe to release it. This application would allow the smart glove to work in a collaborative environment with the robot such as human-robot hand-over tasks in smart manufacturing.

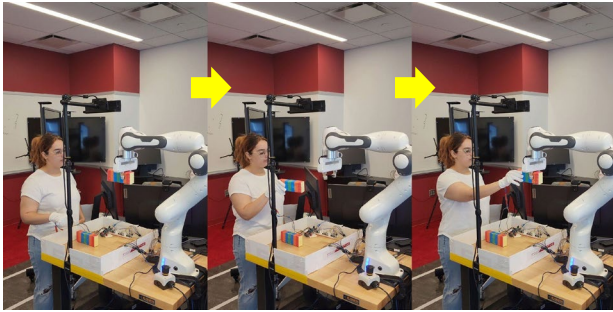


Fig. 9. Receiving an object delivered from the collaborative robot.

After implementing the experiment and testing the three typical cases. We found that the model was reliable in regard to detecting the pinch gesture and controlling the robot was simple and straightforward. However, we noticed that using the smart glove with the robotic arm can lead to multiple different uncertainties due to using the smart glove as a standalone device. If the smart glove was implemented alongside other technologies such as a web camera and video processing would lead to a more fluent and accurate interaction in the human-robot team.

VI. CONCLUSIONS AND FUTURE WORK

In this study, we have developed an AI-embedded and ROS-compatible smart glove system to realize real-time human-robot interaction in collaborative tasks. We have proposed an Extreme Learning Machine (ELM)-based human gesture understanding approach using the data from a set of strip and force sensors of the smart glove and effectively run it through ROS. Three typical baseline gestures have been designed and used for ELM training and validation. The developed system and proposed approach have been implemented in real-world human-robot collaborative tasks with efficiency and success. To achieve a higher accuracy of human gesture recognition for human-robot interaction, we will continue to optimize the ELM training with

more collected data from diverse human users in our future work. In addition, we will conduct more user studies to evaluate the performance of the developed smart glove system in human-robot interactive contexts.

ACKNOWLEDGMENT

Laury Rodriguez gratefully acknowledges receiving a summer research stipend from the NSF Garden State-LSAMP project (#1909824). 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-2117308.

REFERENCES

- [1] P. Tsarouchi, S. Makris, and G. Chryssolouris, "Human-robot interaction review and challenges on task planning and programming," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 8, pp. 916-931, 2016.
- [2] 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.
- [3] W. Wang, Y. Chen, R. Li, and Y. Jia, "Learning and Comfort in Human-Robot Interaction: A Review," *Applied Sciences*, vol. 9, no. 23, p. 5152, 2019.
- [4] M. Awais and D. Henrich, "Human-robot collaboration by intention recognition using probabilistic state machines," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, 2010, pp. 75-80.
- [5] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 299-306.
- [6] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, 2014: IEEE, pp. 527-534.
- [7] 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.
- [8] 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, doi: 10.1109/TASE.2021.3074873.
- [9] K. Haratiannejadi, N. E. Fard, and R. R. Selmic, "Smart glove and hand gesture-based control interface for multi-rotor aerial vehicles," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019: IEEE, pp. 1956-1962.
- [10] E. Fujiwara, D. Y. Miyatake, M. F. M. dos Santos, and C. K. Suzuki, "Development of a glove-based optical fiber sensor for applications in human-robot interaction," in *2013 8th*

- [11] A. Williams, B. Sebastian, and P. Ben-Tzvi, "Review and analysis of search, extraction, evacuation, and medical field treatment robots," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3, pp. 401-418, 2019.
- [12] S. E. Salcudean, H. Moradi, D. G. Black, and N. Navab, "Robot-Assisted Medical Imaging: A Review," *Proceedings of the IEEE*, 2022.
- [13] P. Rückert, K. Tracht, W. Herfs, S. Roggendorf, V. Schubert, and M. Schneider, "Consolidation of product lifecycle information within human-robot collaboration for assembly of multi-variant products," *Procedia Manufacturing*, vol. 49, pp. 217-221, 2020.
- [14] W. Wang, R. Li, Z. M. Diekel, Y. Chen, Z. Zhang, and Y. Jia, "Controlling Object Hand-Over in Human-Robot Collaboration Via Natural Wearable Sensing," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 59-71, 2019.
- [15] L. Roche, "Kinaesthetic communication: cooperation and negotiation during one dimensional physical interaction with human or virtual partners," Sorbonne université, 2019.
- [16] L. Cohen, S. Haliyo, M. Chetouani, and S. Régnier, "Intention prediction approach to interact naturally with the microworld," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2014: IEEE, pp. 396-401.
- [17] 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*, 2011, pp. 1986-1993.
- [18] J. Aleotti, V. Micelli, and S. Caselli, "Comfortable robot to human object hand-over," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012: IEEE, pp. 771-776.
- [19] A. Al-Yacoub, Y. Zhao, W. Eaton, Y. M. Goh, and N. Lohse, "Improving human robot collaboration through Force/Torque based learning for object manipulation," *Robotics and Computer-Integrated Manufacturing*, vol. 69, p. 102111, 2021.
- [20] X. Yu, W. He, H. Li, and J. Sun, "Adaptive fuzzy full-state and output-feedback control for uncertain robots with output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6994-7007, 2020.
- [21] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine, "Collective robot reinforcement learning with distributed asynchronous guided policy search," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017: IEEE, pp. 79-86.
- [22] Force-sensitive resistor, [Online]. Available: <https://www.sparkfun.com/products/9375>.
- [23] Spark Fun flex sensor, [Online]. Available: <https://www.sparkfun.com/products/14666>.
- [24] Arduino Mega2560, [Online]. Available: <https://store.arduino.cc/products/arduino-mega-2560-rev3>.
- [25] 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.
- [26] A. Koubaa, *Robot operating system (ROS)*. Springer, 2017.
- [27] 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: IEEE, pp. 523-528.
- [28] G.-B. Huang, "What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle," *Cognitive Computation*, vol. 7, no. 3, pp. 263-278, 2015.
- [29] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18-29, 2015.
- [30] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16, pp. 3056-3062, 2007.
- [31] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489-501, 2006/12/01/ 2006.
- [32] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16, pp. 3460-3468, 2008.
- [33] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, "Extreme learning machine: algorithm, theory and applications," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 103-115, 2015.
- [34] 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.