# Context-Aware Gesture Interpretation for Semi-Autonomous Robot Control: A Step Towards a Gesture Language for Robots

Ana Carolina Estrela[1], Tatiana Oliveira[1], Khansa Rekik[2], Marco Simões[1], Grimaldo Silva[1]

***Abstract*— Over the last decade, significant advances have been made in the field of human robot interaction (HRI), such as collaborative workcells in smart factories and autonomous mobile robots. However, for these systems to achieve widespread societal acceptance, more natural and efficient communication interfaces must be developed. This work proposes a computer vision-based human–robot cooperative system, composed of a set of gestures that, when combined into composite sequences, allow intuitive interaction with the environment. Gesture identification relies on a human pose analysis model based on deep learning, which earned the 1st place in the Flying Robots Demo at RoboCup 2025. The semantic and contextual interpretation of these sequences is conducted by Large Language Models (LLMs). Integration between visual perception and linguistic understanding enables a more expressive and adaptable form of interaction. The developed system achieved a high-fidelity gesture detection model, with average precision and recall of 0.94.**

## I. INTRODUCTION

In order for integration between humans and robotic systems to be truly efficient and safe, it is essential to develop forms of communication that are fast, intuitive, and reliable [1].

In this scenario, the use of gestures as a form of communication emerges as a promising strategy. As with human language, where sign language replaces speech in specific contexts, gestures can offer more direct, natural, and accessible interaction with robotic systems, especially in environments where other forms of input, such as voice, are limited. It should be noted, however, that the use of gestures in certain contexts, such as industrial settings, may initially yield subpar performance due to usability challenges [2]. Among these challenges is the difficulty of learning the mapping between intended commands and their associated gestures. Nevertheless, gesture-based interaction remains a compelling alternative when designed appropriately.

In this context, various approaches have been developed to enable human-robot interaction through gestures. These include wearable devices, such as gloves and bracelets equipped with Inertial Measurement Units (IMUs), which offer high accuracy in motion capture [3]. However, these approaches pose a challenge: limited operator adherence. According to Al-Bayati, many human operators are resistant to using personal protective equipment due to discomfort and the associated psychological impact [4]. Furthermore, other approaches utilize depth sensors, such as those found in Kinect devices [5], yet they face logistical and economic challenges, including high costs and hardware complexity. Although methods based exclusively on 2D RGB images have emerged as a promising alternative, such approaches encounter technical hurdles, particularly in continuous gesture recognition, where variations in speed and the absence of clear boundaries compromise interpretive accuracy [6].

To address these limitations, this work proposes a gesture recognition approach based on two main principles. The first is the use of off-the-shelf RGB cameras for gesture detection, and the second is the adoption of gestures that are naturally associated with specific verbs or actions. With respect to the latter, we deliberately avoid relying solely on gesture detectors readily available in frameworks such as MediaPipe [7], as many of their predefined gestures do not correspond to intuitive, human-understandable commands. Instead, we prioritize gestures whose meaning can be more readily inferred by users, for instance, gestures widely associated with particular actions worldwide. Furthermore, the set of gestures developed require low articular complexity and are ,therefore, easy to execute . Although the proposed solution can be applied across different domains, in this work it is applied in a conceptual human-robot smart assembly scenario, where a person interacts with a robot within a workspace, similar to that explored in [2].

The proposed architecture is organized into four main modules: object detection and keypoint extraction, gesture classification, context object detection, and semantic interpretation. Together, these modules enable the system to detect relevant visual elements, recognize gestural inputs and associated objects, and assign semantic meaning to gesture sequences, ultimately translating them into executable action plans. During the Robocup 2025 international robotics competition, a pilot version of the gesture detector proposed in this work was tested with the aim of evaluating its ability to promote human-robot interaction, being decisive for the first place won by the team BahiaRT [8].

## II. LITERATURE REVIEW

### A. Natural gestures and intuitive command design

Gestures are a fundamental component of human communication and are closely related to how people express and interpret meaning [9]. Studies in gesture research also

[1] These authors are with Universidade do Estado da Bahia (UNEB), Rua Silveira Martins, 2555, Cabula, Salvador–BA, Brasil, CEP 41150-000. {anacarolgestrela@gmail.com, tatianarrr88@gmail.com, msimoes@uneb.br, jose.jgrimaldo@gmail.com}

[2]Khansa Rekik is with the Department of Systems Engineering, ZeMA gGmbH, Saarbrücken, Germany khansa.rekik@zema.de

The work is a part of the RICAIP project funded the European Union's Horizon 2020 research and innovation programme under grant agreement No 857306

indicate that some gestural functions, such as pointing and head movements associated with affirmation or negation, recur across cultures even when their exact form varies [10]. For the present work, this is important because the objective is not only to detect hand configurations, but also to define a gesture vocabulary whose meaning can be inferred with little prior training. Therefore, gestures with lower articulatory complexity and clearer semantic association are more appropriate for human-robot interaction in assembly environments.

### B. Gesture recognition for practical human-robot interaction

Existing gesture-recognition systems differ mainly in the sensing modality they rely on. Some approaches use wearable sensing and can achieve robust recognition from body signals, but they require the operator to wear dedicated devices [11]. Other approaches rely on Kinect-like systems that combine depth and color information, which can improve robustness but depend on specific sensing hardware [12]. A third line of work explores recognition directly from RGB visual data, seeking lower-cost and more scalable solutions with conventional cameras [13]. For an industrial interaction system intended to be simple to deploy, the RGB-only alternative is attractive because it reduces instrumentation demands and avoids reliance on equipment attached to the operator, which may affect user adherence [4].

### C. Beyond isolated gestures: context and interpretation

Although gesture recognition is a central component of human-robot interaction, recognizing isolated gestures is not sufficient for collaborative task execution. In practical assembly scenarios, the system must also determine which object is being referenced and how a sequence of gestures should be interpreted as a task request. Recent work on multimodal interaction for collaborative assembly reinforces this need for higher-level interpretation, showing that robot assistance depends on combining user signals with contextual task information [2]. In the same direction, multimodal reference-resolution studies show that deictic behavior such as pointing must be interpreted together with environmental cues in order to correctly identify target objects [14].

Thus, the contribution of the present work is not limited to gesture classification itself, but lies in combining low-cost visual recognition with context-aware gesture composition and semantic interpretation for semi-autonomous robot control.

## III. GESTURE RECOGNITION

This section presents an approach for hand gesture detection that combines computer vision techniques with neural network classification, its role in the system architecture is shown in Fig. 1. The method first identifies hand keypoints from 2D RGB images and then uses these keypoints to recognize associated gestures. The selected gestures were chosen for their interpretability, allowing a clearer correspondence between each gesture and the resulting robot action.

### A. Feature Extraction from Body Keypoints

Detecting keypoints was performed using a YOLOv8n-pose [15] that was fine-tuned with hand keypoint data by using the public HaGRID dataset (Hand Gesture Recognition Image Dataset) [16]. The customized YOLOv8n-pose outputs the spatial coordinates $(x, y)$ of each detected joint point. After the detection phase, a post-processing pipeline extracts 185 features from 20 keypoints using a geometric approach. In particular, for pairs of hand keypoints, the method computes Euclidean distances as well as angular descriptors represented through sine and cosine terms.

Let each detected keypoint be denoted by

$$p_i = (x_i, y_i), \quad i \in \{1, \ldots, 20\}. \tag{1}$$

Given two keypoints $p_i$ and $p_j$, the Euclidean distance between them is defined as

$$d_{ij} = \|p_j - p_i\|_2 = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \tag{2}$$

The relative direction from $p_i$ to $p_j$ can be represented by the vector

$$v_{ij} = p_j - p_i = (x_j - x_i,\ y_j - y_i). \tag{3}$$

From this vector, angular information is encoded through its normalized sine and cosine components:

$$\cos\theta_{ij} = \frac{x_j - x_i}{d_{ij}}, \tag{4}$$

$$\sin\theta_{ij} = \frac{y_j - y_i}{d_{ij}}. \tag{5}$$

Thus, each pair of keypoints $(p_i, p_j)$ can be described by the feature set

$$f_{ij} = [d_{ij},\ \cos\theta_{ij},\ \sin\theta_{ij}]\,. \tag{6}$$

To keep the feature vector lightweight, some keypoint pairs were excluded, particularly those whose relative positions are largely invariant, such as the nose and eye keypoints.

In addition, specific characteristics of the thumb orientation are extracted by calculating the axial angle between the wrist and the tip of the thumb, as well as the relative distance between the wrist and the tip of the index finger. Such measurements allow quantifying the manual opening and directionality of the gesture, incorporating information both in a global way and in relative orientation, which enriches the vector of features for tasks of classification of signals and gestures in computer vision.

In summary, this representation captures both the static spatial configuration and the directional relationships between segments.

### B. Gesture classification module

The set of gestures was defined by prioritizing gestures with low articulatory complexity, ease of execution, and semantically intuitive interpretation with wide intercultural recognition. Based on these criteria, the selected gestures were pointing, thumbs-up, thumbs-down, numbers one to five, and holding, as presented in Table I.

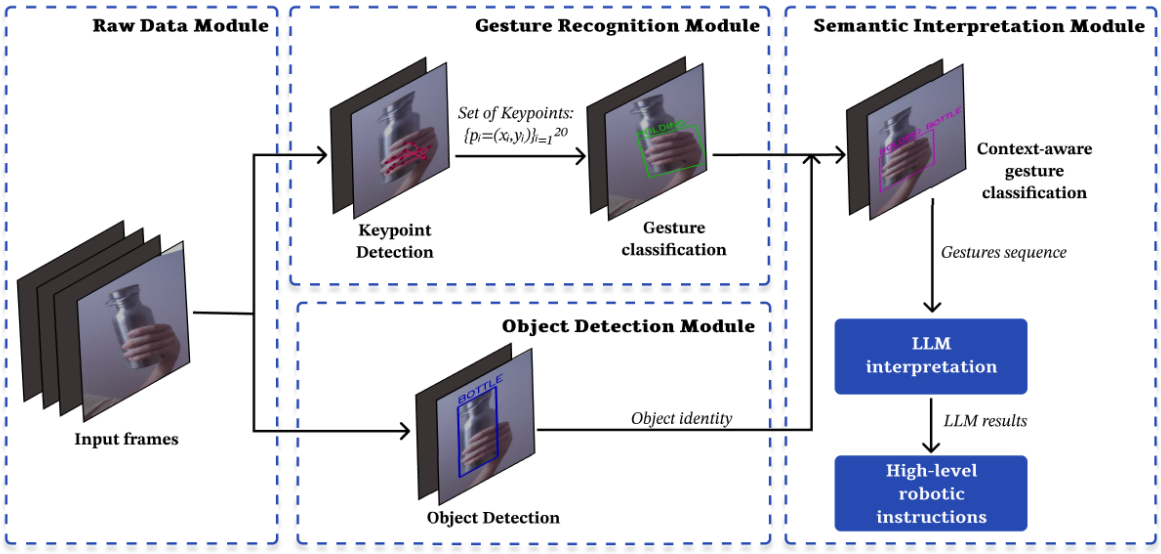



Fig. 1: Architecture of the proposed system.

Our goal is then to train a detector that recognizes, based on the feature set extracted in Sec. III-A, the gestures shown by a person in real-time. To that end, a novel dataset was created including 6 participants and 6418 images between gestures and non-gestures and an average of 641 gesture samples per class.

Finally, based on the dataset and its features, PyTorch [17] was used to train a deep neural network to classify gestures from extracted keypoints.

## IV. CONTEXT-AWARE GESTURE COMPOSITION

To allow complex interactions with the environment, a sequence of gestures might be necessary, and LLMs are implemented in order to interpret those sequences within the context of the environment and the objects present in the scene. Its role in our architecture is shown in Fig. 1.

In that setting, the following structure was developed: The gestures "holding" and "pointing" are used to indicate which object in the scene is the focus of the interaction, as demonstrated in Fig. 2(a) and (b). Pointing is intended for heavy or hard-to-reach objects, in contrast, the holding gesture is more suitable for nearby objects and direct manipulation.

The numbers ("one", "two", "three", "four", and "five") represent the number of units of a certain material that should be brought over the worktable. The remaining gestures, "positive" and "negative", can be customized to different conditions.

In this work, a simulated smart assembly quality control was used as an application to validate the system. In this context, the "positive" gesture is used to signal that the manufacturing process was completed successfully, allowing the object to be moved to the transport area. In contrast, the "negative" gesture is used to signal that the manufacturing process failed and the object should be transported to the waste area.

### A. Context-Aware Gestures

In order to incorporate object detection in the system, a pre-trained YOLO object detection model trained on the COCO (Common Objects in Context) dataset [18] was used. Although most of our experiments were done using a single object "bottle", any supported object would generally be compatible.

Object intersection is done by evaluating hand and object bounding boxes are sufficiently close and intercept. Although pointing could also be performed without bounding box contact, by projecting the pointing direction, in practice this alternative proved difficult to calibrate reliably due to the lack of depth perception.

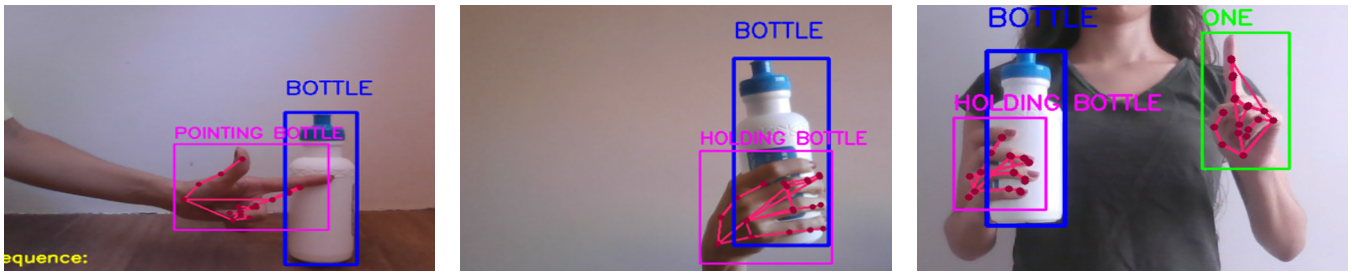
The system implements a mechanism of sequential compound gestures to form complex commands from detected basic gestures. The composition logic operates as follows:

- After the detection of one or more gestures, a time window of 5 seconds begins
- In case no subsequent gesture is detected within this period, the sequence is closed and sent to processing
- The timer resets with each new gesture detected in the sequence

### B. Semantic Interpretation Module

The semantic interpretation module is responsible for converting recognized gesture sequences and their interaction with detected objects into executable robotic actions and also provides another layer of context about the environment.

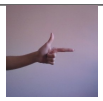
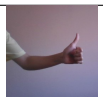
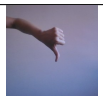
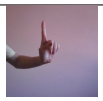
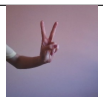
While the perception modules extract structured information such as gesture labels and object identities, these signals alone do not fully specify the intended task. An additional reasoning layer is therefore required to infer the operator's intent and translate it into a coherent action plan. In this work, an LLM is used as a lightweight reasoning component that maps gesture sequences into high-level robotic instructions. The model receives as input a symbolic representation of the




(a) Gesture-Object interaction: Pointing (b) Gesture-Object interaction: Holding (c) Multiple gestures and object interaction

Fig. 2: Gesture interacting with objects

TABLE I: Description of Hand Gestures

| Gesture | Description | Image |
|---|---|---|
| Pointing | Index finger and thumb extended, other fingers flexed |  |
| Positive | Closed fist with thumb extended upwards |  |
| Negative | Closed fist with thumb extended down |  |
| Number 1 | Closed hand with extended index finger |  |
| Number 2 | Closed hand with index and middle fingers extended |  |
| Number 3 | Closed hand with extended index, middle, and ring fingers |  |
| Number 4 | Open hand with four fingers extended and thumb flexed |  |
| Number 5 | Hand completely open with all fingers extended |  |
| Holding | Fully closed hand |  |

detected gestures and relevant objects and generates a concise description of the actions to be executed by a robot. The generated commands are dependent system prompt, which provides context on the robot and the task at hand, thus, the LLM enables flexible interpretation of gesture combinations that can be customized without direct programming.

To guide the interpretation process, a small set of representative gesture-action pairs is provided during initialization. For instance, sequences combining holding an object with a numerical gesture are interpreted as requests to retrieve a given quantity of that object from storage and deliver it to the worktable. Similarly, sequences holding an object while gesturing thumbs-down should be interpreted as a command to remove the object from the worktable and move it to a designated discard area. These examples act as semantic anchors that help the model infer consistent instructions while still allowing some degree of generalization.

For deployment, the semantic interpretation module is executed locally using the Ollama framework [19], which enables the integration of LLMs within the robotic pipeline without requiring external cloud services.

## V. EXPERIMENTAL VALIDATION

All records for the tests were captured at a fixed resolution of 720p, establishing technical uniformity in the image parameters. Variability was systematically introduced by positioning the participant in different coordinates of a $2m^2$ space, covering the entire area where the hand remained visible to the camera. The experiment included radial displacements of the approach and distance in relation to the capture device, resulting in a dynamic distance that varied between 50 cm (maximum approach) and 2 meters (maximum distance) between the operator and the camera. To optimize processing and minimize temporal redundancies, frame extraction was performed every 10 frames.

The hardware configuration of the setup consisted of an NVIDIA GTX 1650 GPU and an Intel i5 CPU.

### A. Gesture detection metrics

The single gesture detection model was trained for 300 epochs, using an authorial dataset composed of 6418 pictures. The hyperparameter combinations shown in Table II were tested.

TABLE II: Hyperparameter Comparison and Model Performance. Best model in bold.

| Learning Rate | Weight Decay | Final Loss | Final Accuracy |
|---|---|---|---|
| **0.0005** | $\mathbf{1 \times 10^{-6}}$ | **0.0645** | **0.9752** |
| 0.0100 | $1 \times 10^{-6}$ | 0.1925 | 0.9357 |
| 0.0100 | $1 \times 10^{-5}$ | 0.3125 | 0.8977 |
| 0.0005 | $1 \times 10^{-5}$ | 0.0708 | 0.9730 |

As illustrated in Table II, the optimal hyperparameter configuration consisted of the minimum values for both

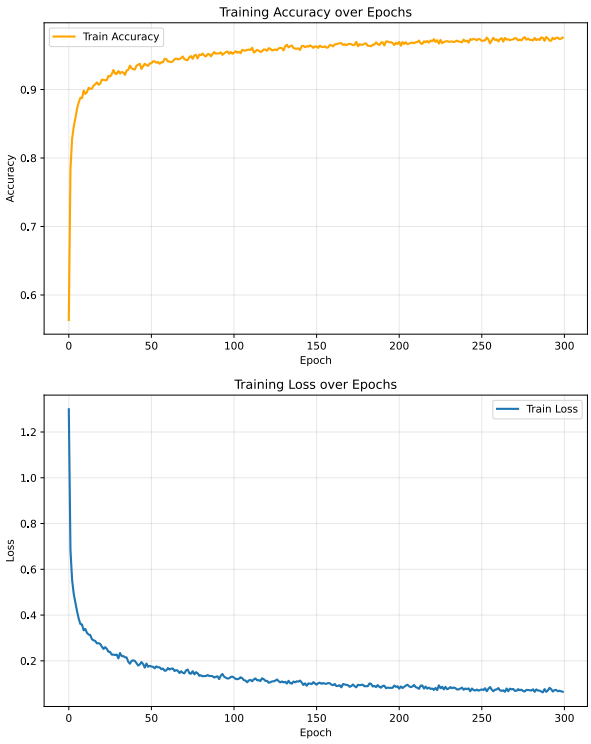



Fig. 3: Loss and Accuracy Results for Best Model

the learning rate and the weight decay, respectively being 0.0005 and $1 \times 10^{-6}$. This combination allowed the model to improve its performance without large oscillations in loss or accuracy, as shown in Figure 3, which indicates the expected inverse relationship between the two metrics.

The average inference time is 1.15 ms. This low latency ensures that the gesture classification step adds minimal overhead to the total processing time of the system.

Table III presents model performance metrics by class, including precision, recall, and F1-score, as well as the support of each class. Table III shows that the model presented high performance in all classes, with precision, recall, and F1-score higher than 0.9 for most gestures, especially classes holding (0.96) and negative (0.97). More challenging classes, such as two and pointing, achieved slightly lower F1-scores, which may be related to visual similarity with other gestures.

### B. *Interpretability testing*

One of our goals is to create gesture sequences that are intuitive for both a non-expert human user and an LLM. Thus, tests to validate that premise were conducted.

Volunteers were selected from university students in the exact sciences ($n = 19$). Of these, 13 identified as male, 1 as female, and the remainder preferred not to answer. Volunteers were between 18 and 25 years old and had no prior information about our approach but had some prior technical knowledge with computers.

Before the survey began, participants were shown a tutorial example containing a gesture sequence and its intended action in a smart assembly scenario. Afterwards, a self-

TABLE III: Metrics by Gesture Classification Result

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| pointing | 0.93 | 0.92 | 0.92 | 148 |
| holding | 0.96 | 0.97 | 0.96 | 93 |
| positive | 0.92 | 0.96 | 0.94 | 158 |
| negative | 1.00 | 0.95 | 0.97 | 110 |
| one | 0.93 | 0.93 | 0.93 | 140 |
| two | 0.93 | 0.91 | 0.92 | 172 |
| three | 0.96 | 0.93 | 0.95 | 138 |
| four | 0.92 | 0.97 | 0.95 | 115 |
| five | 0.95 | 0.95 | 0.95 | 110 |
| non_action | 0.90 | 0.91 | 0.91 | 100 |
| accuracy | | | 0.94 | 1284 |
| macro avg | 0.94 | 0.94 | 0.94 | 1284 |
| weighted avg | 0.94 | 0.94 | 0.94 | 1284 |

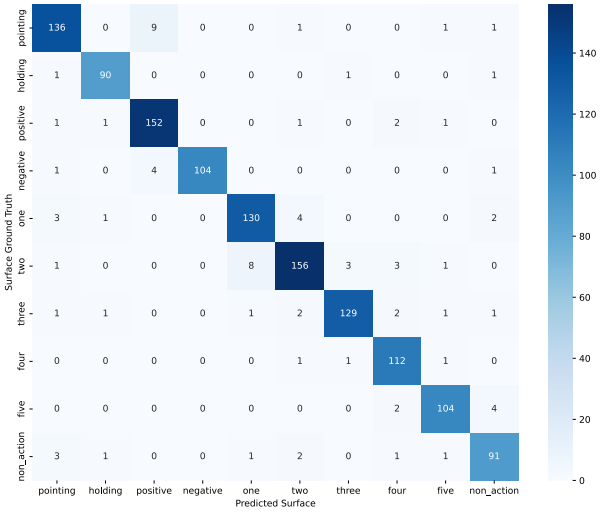



Fig. 4: Confusion Matrix

contained survey with 4 gesture sequence challenges was provided. Users were then tasked with associating, by themselves, some gesture sequences with the desired action - including novel sequences that were not presented in the tutorial.

The same scenarios were put to the test with the proposed semantic interpretation LLM module configured to translate input gestures into executable commands. As seen in Table IV, the majority volunteers reached consensus on the intended interpretation even in scenarios where no prior context was provided, such as in Q4.

It is noteworthy that the system inference mirrored this human behavior, achieving consistent results. The mismatches observed in Q1 for the $T = 0.5$ $T = 0.8$ configuration likely stem from an LLM hallucination, which is a characteristic high and moderate temperature setting in generative models. At the end of the survey, users were asked to suggest other gestures that could be added to the vocabulary. About 32% felt that an emergency stop gesture (open hand for robot to stop) would be important. Moreover, nearly 20% of participants would like clearer ways of specifying higher number (10+number for example). Finally, 8 % of participants thought that being able to summon the robot to

TABLE IV: Human-LLM Alignment in Gesture Sequence Interpretation

| Q. | Target Interpretation | Human Consensus | System Accuracy (% Match) T=0.1 | T=0.5 | T=0.8 |
|---|---|---|---|---|---|
| Q1 | Bring identical item | 89.5% | 94.7% | 73.7% | 79.0% |
| Q2 | Discard product | 78.9% | 100.0% | 100.0% | 100.0% |
| Q3 | Fetch 3 units | 100.0% | 100.0% | 100.0% | 100.0% |
| Q4 | Bring 1 unit | 68.4% | 100.0% | 100.0% | 100.0% |

*Note:* Human consensus and LLM accuracy are both computed over $n = 19$ responses per question.
$T$ denotes the model's temperature parameter. Accuracy reflects the percentage of successful matches.

the worktable would be useful, e.g. a hand moving repeatedly backward. Other answers did not converge (singletons) or the participant had no suggestion.

## VI. CONCLUSIONS

The proposed system enables natural gesture-based communication between humans and autonomous agents. Its fundamental advantage lies in the low implementation cost, achieved through the use of a single RGB camera, locally deployed and free LLMs, and a modular architecture that spans from gesture recognition to autonomous action planning.

The feasibility and effectiveness of the system were validated through gesture-classification experiments, which achieved an average accuracy of $94\%$. The semantic interpretation study also showed that the majority of participants with no prior knowledge of the system could consistently interpret the gesture sequences.

Despite the positive outcomes achieved, this research opens opportunities for future work. A key direction involves extending the proposed system to support an organically expanding gesture space, enabling the integration of new commands, behaviors, and interaction patterns over time.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to Caio Lucena Andrade Paula, Gabrielle Ferreira de Souza Carvalho, Reinaldo da Silva Junior, and Vitória Matos, who provided significant assistance during the gesture dataset composition. Their support was fundamental for the creation of a diverse dataset.

## REFERENCES

[1] A. Buerkle, W. Eaton, A. Al-Yacoub, M. Zimmer, P. Kinnell, M. Henshaw, M. Coombes, W.-H. Chen, and N. Lohse, "Towards industrial robots as a service: Flexibility, usability, safety and business models," *Robotics and Computer-Integrated Manufacturing*, vol. 81, 2023.

[2] K. Rekik, G. Silva, A. Bashir, and R. Müller, "Multimodal interaction for human-robot collaboration in assembly: An llm-enhanced approach," in *2025 IEEE 21st Int. Conf. on Automation Science and Engineering (CASE)*. IEEE, 2025, pp. 1207–1212.

[3] I. Kvasić, D. O. Antillon, Đ. NAĐ, C. Walker, I. Anderson, and N. Mišković, "Diver-robot communication dataset for underwater hand gesture recognition," *Computer Networks*, vol. 245, 2024.

[4] A. J. Al-Bayati, A. T. Rener, M. P. Listello, and M. Mohamed, "Ppe non-compliance among construction workers: an assessment of contributing factors utilizing fuzzy theory," *Journal of Safety Research*, vol. 85, pp. 129–140, 2023.

[5] Y. Shi, Y. Li, X. Fu, M. Kaibin, and M. Qiguang, "Review of dynamic gesture recognition," *Virtual Reality & Intelligent Hardware*, vol. 3, pp. 183–206, 2021.

[6] R. Mahmoud, S. Belgacem, and M. N. Omri, "Deep signature-based isolated and large scale continuous gesture recognition approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1793–1807, 2022.

[7] C. Lugaresi, F. Tang, M. Nash, C. McClanahan, and et al., "Mediapipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[8] RoboCup Brasil, "Flying robots league: Flying robot demo na robocup 2025," https://robocup.org.br/wiki/doku.php?id=flying, 2025, accessed: 2026-05-22.

[9] S. Goldin-Meadow and M. W. Alibali, "Gesture's role in speaking, learning, and creating language," *Annual Review of Psychology*, 2013.

[10] N. Abner, K. Cooperrider, and Goldin-Meadow, "Gesture for linguists: a handy primer," *Language and Linguistics Compass*, vol. 9, no. 11, pp. 437–451, 2015.

[11] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 293–299, 2014.

[12] Z. Ren, J. Meng, J. Yuan, and Z. Zhang, "Robust hand gesture recognition with kinect sensor," 11 2011, pp. 759–760.

[13] C. C. dos Santos, J. L. A. Samatelo, and R. F. Vassallo, "Dynamic gesture recognition by using cnns and star rgb: A temporal information condensation," *Neurocomputing*, vol. 400, pp. 238–254, 2020.

[14] D. Kontogiorgos, E. Sibirtseva, A. Pereira, G. Skantze, and J. Gustafson, "Multimodal reference resolution in collaborative assembly tasks," in *Proceedings of the 4th International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction*, 2018, pp. 38–42.

[15] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2022, pp. 2637–2646.

[16] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk, "Hagrid–hand gesture recognition image dataset," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4572–4581.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[18] R. Varghese and S. M., "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024, pp. 1–6.

[19] F. S. Marcondes, A. Gala, R. Magalh aes, F. Perez de Britto, D. Dur aes, and P. Novais, *Using Ollama*. Cham: Springer Nature Switzerland, 2025, pp. 23–35. [Online]. Available: https://doi.org/10.1007/978-3-031-76631-2_3