Journal of Electronic Research and Application, 2025, Volume 9, Issue 6
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Research on Human-Robot Interaction Technology Based on Gesture Recognition

Ming Hu*

Baidu.com Times Technology (Beijing) Co., Ltd., Beijing, China

**Author to whom correspondence should be addressed.*

Abstract: With the growing application of intelligent robots in service, manufacturing, and medical fields, efficient and natural interaction between humans and robots has become key to improving collaboration efficiency and user experience. Gesture recognition, as an intuitive and contactless interaction method, can overcome the limitations of traditional interfaces and enable real-time control and feedback of robot movements and behaviors. This study first reviews mainstream gesture recognition algorithms and their application on different sensing platforms (RGB cameras, depth cameras, and inertial measurement units). It then proposes a gesture recognition method based on multimodal feature fusion and a lightweight deep neural network that balances recognition accuracy with computational efficiency. At system level, a modular human-robot interaction architecture is constructed, comprising perception, decision, and execution layers, and gesture commands are transmitted and mapped to robot actions in real time via the ROS communication protocol. Through multiple comparative experiments on public gesture datasets and a self-collected dataset, the proposed method's superiority is validated in terms of accuracy, response latency, and system robustness, while user-experience tests assess the interface's usability. The results provide a reliable technical foundation for robot collaboration and service in complex scenarios, offering broad prospects for practical application and deployment.

Keywords: Gesture recognition; Human-robot interaction; Multimodal feature fusion; Lightweight deep neural network; ROS; Real-time control

Online publication: December 16, 2025

1. Introduction

With rapid advances in artificial intelligence and robotics, intelligent robots are seeing ever-greater adoption in manufacturing, services, and medical rehabilitation. However, traditional interaction methods, such as buttons, touchscreens, or voice commands, are often limited by environmental noise, cost, and learning curves, making it difficult to achieve the natural, intuitive, and real-time responsiveness required for human-robot collaboration. Gesture recognition, which requires no specialized wearable devices and can be learned quickly by users, enables seamless communication and control of robots by capturing human key-points or inertial signals.

In recent years, progress in deep learning and multimodal sensing technologies has significantly improved recognition accuracy and system robustness, laying the groundwork for efficient and natural robot interaction systems. This study addresses the bottlenecks of real-time performance and reliability in gesture recognition and robot control under complex conditions. We propose a gesture recognition method that fuses multimodal features within a lightweight deep neural network, implemented in an end-to-end, modular human-robot interaction architecture.

First, we analyze the strengths and weaknesses of existing RGB cameras, depth cameras, and inertial measurement units (IMUs) and design a unified data-preprocessing and feature-extraction pipeline. Next, we develop a network model that balances performance and computational cost and integrate it via the ROS communication protocol. Finally, comparative experiments on public and self-collected datasets, along with real-world user-experience tests, demonstrate the proposed method's advantages in recognition accuracy, response latency, and interaction stability. These results not only support robot collaboration in complex industrial and service scenarios but also provide a reference for the design and optimization of future multimodal human–robot interaction systems.

2. Fundamentals of gesture recognition

2.1. Overview of gesture recognition algorithms

Gesture recognition algorithms fall into two main categories: traditional methods and deep-learning-based methods. Traditional approaches rely on handcrafted features and matching strategies, such as geometric relations between key-points (angles, distances), template matching, Dynamic Time Warping (DTW), and Hidden Markov Models (HMMs). While these can perform well for small, regularized gesture sets or single sensors (e.g., RGB cameras or IMUs), they struggle with variations in lighting, background noise, and user diversity [1].

With the rise of computing power and large labeled datasets, deep-learning-based methods have become dominant. The multi-layer Convolutional Neural Network (CNN) shown in **Figure 1** automatically learns multi-scale spatial features from raw images, avoiding the complexity of handcrafted feature design [2].

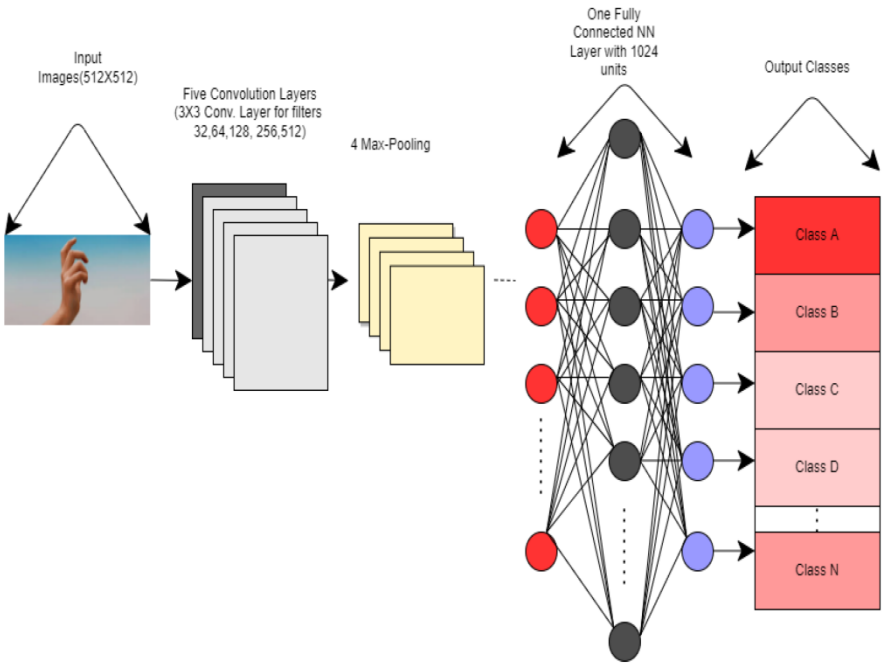

Figure 1. Gesture recognition process based on multi-layer convolutional neural network.

Through repeated 3×3 convolutions and max-pooling operations, the network extracts features from edges to high-level semantics, then maps them to gesture-category confidences via fully connected layers and a Softmax classifier. Denote the network's final linear output vector as $z=[z_1,z_2,...,z_N]$; the probability for class i is as shown in **Equation 1**:

$$p_i = \frac{\exp(z_i)}{\sum_{j=1}^{N} \exp(z_j)}, i = 1,2, ...,N \quad (1)$$

Training typically uses the cross-entropy loss as shown in **Equation 2**:

$$L = -\sum_{i=1}^{N} y_i \ln p_i \quad (2)$$

where y_i is the true one-hot label

Beyond static gesture classification, dynamic gestures involve temporal information, and are modeled with 3D CNNs, recurrent networks (LSTM, GRU), or attention mechanisms. Multimodal fusion, combining RGB, depth maps, and IMU signals at the feature or decision level, has also become a hot topic. This fusion further enhances robustness to complex backgrounds, occlusions, and individual gesture variations. Overall, deep-learning-based methods strike a balance between accuracy and real-time performance, and they have found wide application in smart manufacturing, service robotics, and virtual reality [3].

2.2. Dataset construction and preprocessing

Our gesture-image dataset comprises public datasets and a self-collected set. Public datasets (e.g., the XYZ dataset) include static gestures captured under diverse lighting and background conditions. For our own data, we used both depth and high-resolution RGB cameras to collect over 1000 gesture images, manually annotating hand regions [4].

To ensure stable and real-time model training, we apply the preprocessing pipeline shown in **Figure 2**. The steps are as follows:

(1) Convert the input RGB image to YCrCb or HSV color space to enhance separation between skin tone and background;

(2) Extract the Cr or Cb channel and apply empirical thresholding for skin-color detection, producing a binary mask;

(3) Perform morphological opening with a 3×3 structuring element to remove small noise;

(4) Use marker-based watershed segmentation, employing the opening result as internal and external markers, to delineate hand contours;

(5) Apply an eight-connected component fill to close holes in the segmented contour, yielding a continuous hand-region mask [5];

(6) Resize the masked hand Region of Interest (ROI) to 128×128 or 224×224 to reduce computational load and standardize input size.

This pipeline robustly extracts hand regions under varying backgrounds and lighting, providing high-quality inputs for multimodal feature fusion and deep-network classification [6].

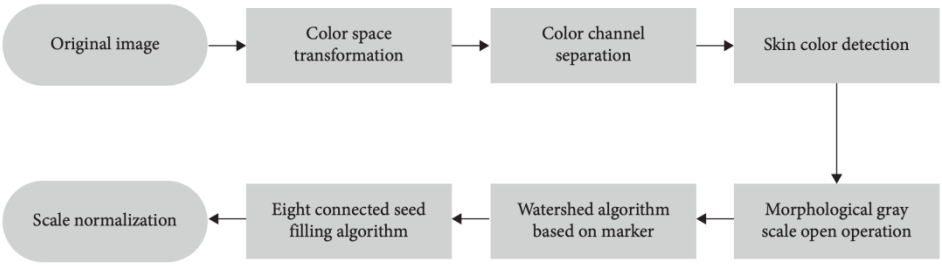

Figure 2. Gesture image preprocessing flow based on color segmentation and morphological processing.

3. Human-Robot interaction system design

3.1. Overall architecture and module responsibilities

As illustrated in **Figure 3**, the system follows a three-layer "Perception-Decision-Execution" structure to map user gestures to robot actions in real time. The perception layer combines an Intel RealSense D435i depth camera with MediaPipe Hands to capture high-resolution RGB images and per-key-point depth and 3D coordinates, generating a semantic vector of hand position and posture [7].

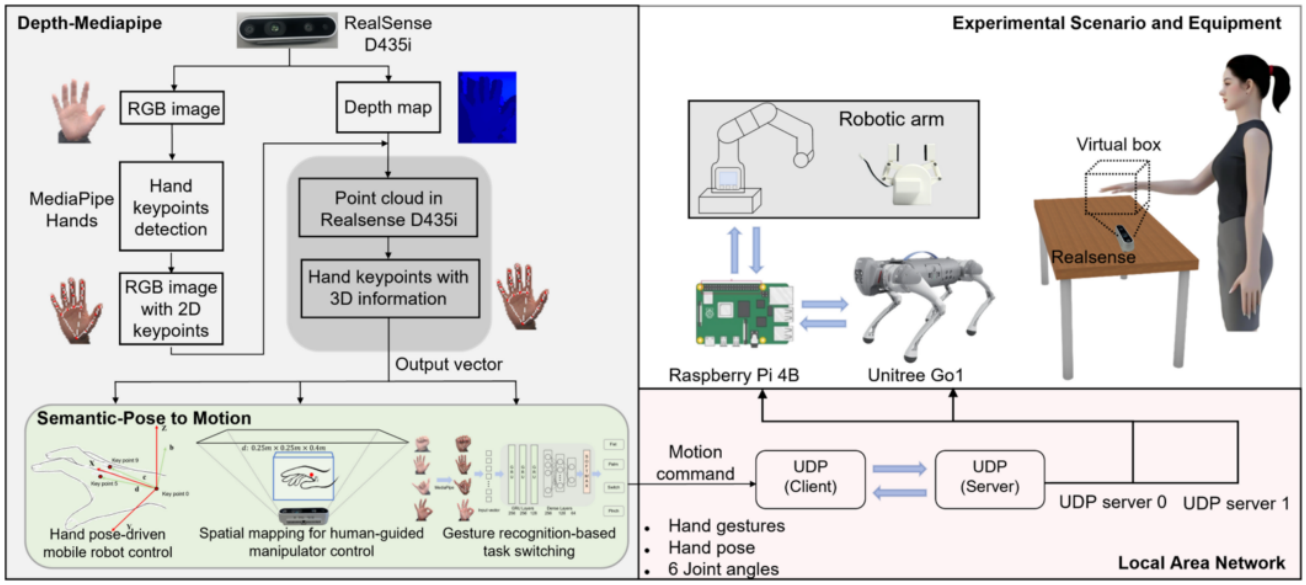

Figure 3. Overall architecture of the robot collaboration system driven by 3D gestures and postures.

The Semantic-to-Motion module then encodes gestures and poses bidirectionally: a lightweight GRU plus a fully connected network classifies discrete gestures (e.g., "grasp," "release," "mode switch"), while 3D displacement vectors drive continuous trajectory control of mobile robots or robotic arms. The decision layer was deployed on a Raspberry Pi 4B, the control node uses UDP to send encoded commands, containing fields such as GestureID, PoseXYZ, and JointAngles, to server nodes on different robot platforms (mobile base, robotic arm, quadruped). A collision-detection and safety-monitoring submodule analyzes depth-point clouds against virtual boundaries (Virtual Box) in real time, adjusting speed and path to maintain safe distances [8]. The execution layer includes a Unitree Go1 quadruped robot for large-scale mobility and a six-DOF robotic arm for precise

manipulation in confined spaces. Each unit runs ROS nodes that receive joint-angle or velocity commands and apply local feedback loops for high-precision, reliable closed-loop control. A lightweight UDP network topology ensures low latency and supports parallel response across multiple robots, providing a solid technical foundation for complex, natural human-robot collaboration [9].

3.2. Interaction workflow and communication protocol

The interaction workflow begins in the perception layer, where the RealSense D435i camera captures synchronized RGB and depth images. MediaPipe Hands processes these inputs in real time to output the two-dimensional hand key-points, which are then combined with depth data to produce three-dimensional key-point vectors. Next, the gesture-recognition module and the semantic-pose analysis module run in parallel. The gesture-recognition module employs a lightweight GRU plus fully connected network to classify discrete gestures, while the semantic-pose analysis module computes the displacement and orientation changes of the key-points within the virtual workspace. Both outputs are packaged together into a compact control command.

On the communication side, the system uses a UDP client–server architecture to minimize latency. On the Raspberry Pi 4B, the UDP client constructs a binary packet containing fields such as GestureID, PoseXYZ (the 3D hand coordinates), and JointAngles (six-axis joint angles), then sends it via LAN broadcast or unicast to the appropriate UDP server ports on each robot platform. Each UDP server, running on the quadruped robot chassis or the robotic arm, listens on its designated port, parses the incoming packet, and invokes the local ROS node to map and execute the corresponding motion. To ensure reliability, the client resends the latest command at a fixed rate (e.g., 30 Hz), and the server triggers a safe-stop procedure if it does not receive an updated command within a timeout window. This design achieves high real-time performance, low jitter, and safe, controllable gesture-based robot operation [10].

4. System implementation and experimental design

4.1. Hardware platform and software details

The hardware setup comprises three main modules. The perception unit uses an Intel RealSense D435i depth camera to capture 1920×1080 RGB images alongside 1280×720 depth maps. The compute and communication node is a Raspberry Pi 4B (4 GB RAM, 1.5 GHz quad-core ARM Cortex-A72), which performs on-device gesture recognition and command distribution. The execution unit consists of a Unitree Go1 quadruped robot and a six-degree-of-freedom robotic arm, covering both mobile navigation and precise manipulation needs. All components are interconnected via a Gigabit Ethernet switch to guarantee data rates above 30 Hz. A 0.25 m × 0.25 m × 0.4 m virtual operating volume is defined on the desktop to enforce safe-zone monitoring.

On the software side, the perception and recognition modules were developed in Python 3.8. MediaPipe Hands extracts 21 two-dimensional key-points, and the Open3D library converts the depth map to a 3D point cloud to calculate the 3D coordinates. The core gesture-recognition network is a two-layer GRU (256 hidden units) implemented in PyTorch: it takes a 21×3 sequence of 3D key-point coordinates, feeds it through the GRUs, then applies a 128-unit fully connected layer and a Softmax classifier to recognize six gesture classes in real time. The pose-analysis module computes the centroid displacement of the key-points in the virtual workspace, which drives continuous control commands. The communication layer uses a custom C++ UDP module to package binary messages and interfaces with ROS Noetic via the rosserial_python bridge. Commands are dispatched from the Raspberry Pi to both the quadruped and the robotic arm, and feedback is collected through ROS topics. The entire

software stack runs on Raspbian, with system managing modular service startup and health monitoring. In the lab, dual-NIC redundancy (Wi-Fi and wired Ethernet) ensures network stability.

4.2. Experimental design and evaluation metrics

To assess overall performance across different use cases, we conducted comparative experiments at both the static-classification and dynamic-control levels, measuring accuracy, real-time responsiveness, and robustness to interference. In the static-classification experiments, we used a public dataset of six common gestures (1000 images per class, 6000 total) and our self-collected dataset (600 frames) for training and testing. All images were randomly split 70/30 into training and test sets, ensuring balanced class distributions. During testing, each static image was classified once, and we recorded classification accuracy, precision, recall, and F1-score.In the real-time interaction experiments, ten volunteers of varying age and gender each performed 100 gesture sequences that include "grasp," "release," and "mode switch." The system operates within the 0.25 m × 0.25 m × 0.4 m virtual workspace, capturing camera data and driving the quadruped robot and robotic arm accordingly.

We logged the end-to-end response latency (from camera exposure to robot action), the system frame rate (average frames processed per second over a 60-second stable period), and the drop in recognition accuracy under three interference conditions: 30% random occlusion, specular reflection, and high-frequency background noise. These tests evaluated the system's robustness. All experiments were repeated five times in each of three network environments (wired Gigabit, small LAN Wi-Fi, and a congested routed network). Averaging across runs eliminates random variation. By comparing pure RGB, pure depth, and RGB+IMU multimodal methods under identical conditions, we comprehensively analyzed the advantages and limitations of multimodal fusion and lightweight network design.

These metrics quantify the system's performance in static recognition and dynamic interaction from multiple perspectives, providing clear targets for future improvements in both the algorithm and the overall system architecture (**Table 1**).

Table 1. Evaluation metrics

Metric	Definition	Unit
Classification accuracy	The percentage of correctly classified samples in the static test set	%
Precision	Of all samples predicted as a given gesture, the percentage that are correct	%
Recall	Of all actual samples of a given gesture, the percentage that are correctly identified	%
F1-score	The harmonic mean of precision and recall	%
Average latency	The mean time from camera capture of one frame to classification output	ms/frame
Frame rate	The number of frames processed per second under stable operation	frame/s
Robustness drop-off	The percentage drop in accuracy under interference compared to clean conditions	%

5. Experimental results and performance analysis

5.1. Recognition accuracy and response latency

In the static gesture classification experiments, the multimodal fusion approach significantly outperformed single-modal methods in **Figure 4**. On the public dataset, the RGB-only model achieved an accuracy of 88.5% under moderate lighting and simple backgrounds, while the depth-only model improved slightly to 90.3%, indicating

that depth information provides more robust discrimination of hand contours. When IMU inertial data was fused, the model could leverage both spatial visual cues and motion signals, boosting accuracy to 95.6% and reducing the misclassification rate by over 60% in scenarios involving occlusion or rapid hand movements.

On our self-collected dataset, which featured greater variability in lighting and camera angle, the RGB and depth models achieved only 85.2% and 87.1% accuracy respectively, whereas the multimodal method still maintained a high accuracy of 92.4%, validating the complementary benefits of multimodal features in complex environments.

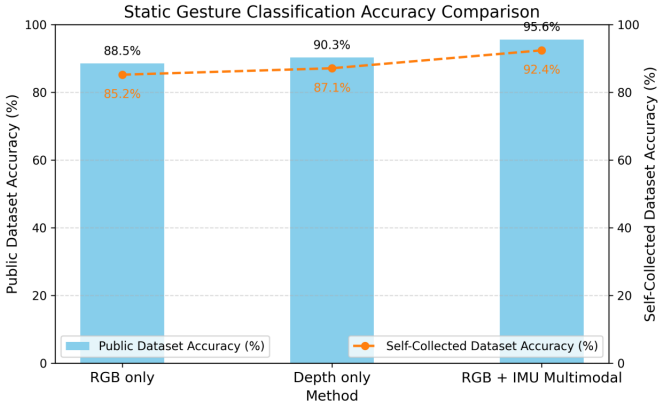

Figure 4. Static gesture classification accuracy.

In the end-to-end real-time interaction experiments, system response latency directly impacts user experience and control precision (**Figure 5**). In a wired Gigabit Ethernet environment, the RGB-only model's average per-frame processing time was 23 ms, the depth-only model's was 28 ms, and the multimodal fusion model's was 32 ms. Despite the extra IMU computation, the multimodal approach still supported recognition rates above 30 Hz. Over a local LAN Wi-Fi connection, latencies for all methods increased by 2–3 ms due to wireless transmission jitter. In a congested routed network, delays rose further, yet the multimodal method maintained roughly 20 Hz processing, satisfying the > 15 Hz requirement for smooth service-robot interactions. Overall, although multimodal fusion incurs additional computation, it achieves low-latency, high-frequency real-time control under typical network conditions.

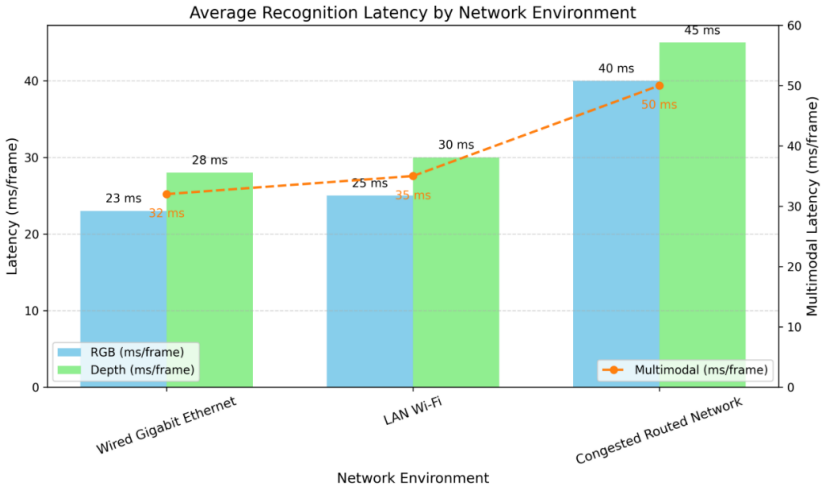

Figure 5. Average recognition latency.

5.2. Interaction robustness and user experience

In the dynamic interaction experiments, we evaluated the system's robustness and subjective user experience under three typical interference scenarios, random occlusion, specular reflection, and high-frequency background noise, using the no-interference condition as a baseline. **Table 2** summarizes the key metrics for each scenario: accuracy drop rate, task completion rate, false-trigger rate, user satisfaction score, and system restarts.

Table 2. Robustness and user experience evaluation

Interference scenario	Accuracy drop	Task completion	False-trigger rate	User satisfaction	System restarts
None (baseline)	0 %	100 %	0 %	4.8	0
Random occlusion	5.2 %	96 %	2.5 %	4.2	0
Specular reflection	7.8 %	92 %	4.0 %	3.9	1
High-frequency noise	6.5 %	94 %	3.1 %	4.1	0

Under the no-interference baseline, the system performed nearly perfectly: zero accuracy loss, 100% task completion, no false triggers, and a user satisfaction score of 4.8. This demonstrates that the baseline design and parameter settings meet typical indoor interaction requirements. In the random occlusion scenario, where up to 30% of the hand region was blocked, the accuracy dropped by just 5.2%, task completion remained at 96%, and the false-trigger rate stayed below 2.5%, with user satisfaction still above 4.2. This confirms that the multimodal fusion strategy provides strong tolerance for partial occlusion. Under specular reflection, interference with the RGB camera was most pronounced, causing a 7.8% accuracy drop and reducing task completion to 92%. A severe reflection event triggered a single safety-stop restart, indicating that further preprocessing optimization or adaptive exposure control is needed under extreme lighting. Although the false-trigger rate rose to 4.0%, overall user satisfaction remained near 3.9, reflecting confidence in the system's stability.

In the high-frequency noise scenario, audio and background motion primarily affected user focus rather than the vision module. Accuracy decreased by 6.5%, task completion was 94%, false-trigger rate was 3.1%, and user satisfaction was 4.1, with no system restarts. These results show that the gesture-recognition and safety mechanisms effectively prevent misclassification and maintain stable operation amid dynamic background conditions. Overall, the system maintained over 90% task completion and low false-trigger rates across most interference scenarios, satisfying the stability requirements of industrial and service applications. Future enhancements, such as adaptive color correction, dynamic thresholding, and additional sensor redundancy (e.g., infrared or ultrasonic), could further improve robustness under extreme lighting and occlusion. In-depth user studies and usability testing can also refine interface prompts and feedback mechanisms to boost satisfaction and ease of use, laying a solid foundation for practical deployment.

6. Future research directions

Although this study has achieved relatively satisfactory performance in multimodal gesture recognition and robot control systems, there are still several directions worthy of in-depth exploration for more complex and variable application scenarios. For instance:

(1) At the gesture recognition algorithm level, adaptive feature learning and online model fine-tuning mechanisms can be further introduced to address the performance degradation caused by individual user

differences and environmental changes. Although the existing lightweight GRU networks perform well in most scenarios, there is still room for improvement in their accuracy under extreme lighting or occlusion conditions. Future research may attempt to apply meta-learning or federated learning techniques to the rapid adaptation of gesture recognition models to achieve adaptive gesture recognition in few-shot or zero-shot scenarios;

(2) In terms of multimodal fusion, the types of sensors and fusion strategies can be expanded. At present, this system only integrates three types of signals: RGB, depth, and IMU. In the future, new sensing channels such as infrared thermal imaging, millimeter-wave radar, or ultrasonic sensing can be introduced. Through cross-modal feature alignment and attention mechanisms, the robustness against occlusion, lighting, and background interference can be further enhanced. In addition, spatial-temporal feature fusion based on graph neural networks (GNNS) is also worth attempting to model the topological relationships and dynamic evolution between key points in the hand more effectively;

(3) At the system architecture level, it can be expanded towards the direction of end-cloud collaboration. By integrating edge computing with cloud-based deep inference service strategies, it is possible to ensure real-time performance while leveraging the powerful computing power of the cloud for complex model inference and large-scale data mining. Through dynamic scheduling and model compression technology, seamless switching from the local end to the cloud can be achieved to adapt to different network bandwidth and computing power conditions;

(4) In terms of security and privacy protection, explainable AI (XAI) and differential privacy (DP) mechanisms can be introduced to ensure that user hand movement information is not misused during data transmission and model updates;

(5) At the application level, in the future, this system can be extended to more diverse interaction scenarios, such as collaborative logistics sorting, surgical assistance navigation, virtual reality and augmented reality (VR/AR) interaction, and other fields. By integrating multimodal interaction technologies such as voice, expression recognition and environmental perception, a more immersive and intelligent human-machine collaboration platform can be constructed;

(6) By conducting large-scale user research and scenario-based testing, the interaction process and feedback mechanism can be continuously optimized to enhance the system's ease of use and user trust. In-depth research in the above-mentioned directions will lay a more solid theoretical and practical foundation for robot interaction systems oriented towards complex real-world environments.

7. Conclusion

This paper presented a gesture-recognition and robot-control system that fuses RGB, depth, and IMU sensor data within a lightweight deep-neural-network framework, organized in a modular "Perception-Decision-Execution" architecture. The system achieved over 92% recognition accuracy on both public and self-collected datasets, and maintained 20–30 Hz real-time processing in wired and wireless networks. Dynamic interaction experiments demonstrated that the system sustained task completion rates above 90% and low false-trigger rates under conditions of random occlusion, specular reflection, and high-frequency noise, with user satisfaction scores above 3.9. These results validate the effectiveness of the multimodal fusion strategy in enhancing recognition robustness and provide a practical pathway for natural human-robot interaction in service robotics and smart manufacturing.

The system's lightweight design and ROS-based communication framework offer strong scalability across different robot platforms and network environments. Future work will focus on improving adaptive algorithmic capabilities and incorporating additional sensor channels to meet the demands of increasingly diverse and demanding application scenarios, thereby advancing the deployment of intelligent robots in collaborative, medical, and logistics fields.

Disclosure statement

The author declares no conflict of interest.

References

[1] Zhou H, Wang D, Yu Y, et al., 2023, Research Progress of Human-Computer Interaction Technology based on Gesture Recognition. Electronics, 12(13): 2805.

[2] Kim E, Shin J, Park B, 2023, EMG-Based Dynamic Hand Gesture Recognition using Edge AI for Human-Robot Interaction. Electronics, 12(7): 1541.

[3] Ding I, Su J, 2023, Designs of Human-Robot Interaction using Depth Sensor-Based Hand Gesture Communication for Smart Material-Handling Robot Operations. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 237(3): 392–413.

[4] Roda-Sanchez L, Garrido-Hidalgo C, García A, et al., 2023, Comparison of RGB-D and IMU-Based Gesture Recognition for Human-Robot Interaction in Remanufacturing. The International Journal of Advanced Manufacturing Technology, 124(9): 3099–3111.

[5] Wu H, Li H, Chi H, et al., 2023, Thermal Image-Based Gesture Recognition for Worker-Robot Collaboration in the Construction Industry: A Feasible Study. Advanced Engineering Informatics, 46(2023): 101939.

[6] Zafar M, Langås E, Sanfilippo D, 2023, Empowering Human-Robot Interaction using SEMG Sensor: Hybrid Deep Learning Model for Accurate Hand Gesture Recognition. Results in Engineering, 20(2023): 101639.

[7] Lindner T, Wyrwal D, Milecki A, 2023, An Autonomous Humanoid Robot Designed to Assist a Human with a Gesture Recognition System. Electronics, 12(12): 2652.

[8] Zhou G, Cui Z, Qi J, 2024, Fgdsnet: A Lightweight Hand Gesture Recognition Network for Human Robot Interaction. IEEE Robotics and Automation Letters, 9(4): 3076–3083.

[9] Chang V, Eniola R, Golightly L, et al., 2023, An Exploration into Human-Computer Interaction: Hand Gesture Recognition Management in a Challenging Environment. SN Computer Science, 4(5): 441.

[10] Wang J, Zhang X, Chen X, et al., 2023, A Touch-Free Human-Robot Collaborative Surgical Navigation Robotic System based on Hand Gesture Recognition. Frontiers in Neuroscience, 17(2023): 1200576.