SEMARAK ILMU PUBLISHING

Semarak Engineering Journal

Journal homepage:
https://semarakilmu.com/
ISSN: 3036-0145



Human-Following System Based on Deep Learning for Supermarket Robot

Wahyudi[1,*], Bonaventura Emmanuel Raditya[1], Yosua Alvin Adi Soetrisno[1]

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Diponegoro, 50275 Semarang, Indonesia

ARTICLE INFO

Article history:
Received 20 February 2026
Received in revised form 15 April 2026
Accepted 16 April 2026
Available online 28 April 2026

Keywords:
RGB Camera; YOLOv11-Pose; Hand Sign; Robot Operating System (ROS)

ABSTRACT

Physical burden and difficulty pushing shopping trolleys have become the issues in supermarket shopping activities, making shopping less convenience. In order to address the issues while promoting customer experience, innovative solutions such as robotics technologies are necessary, eliminating the need to manually push trolleys. Nevertheless, frequently, interaction with robots needs less intuitive methods. Thus, the aim of this research is to design a prototype of supermarket robot. An easy-to-control robot that can follow its user dynamically through hand gestures. Foxy's Robot Operating System (ROS) 2 framework is used to design the robot system, involving RGB camera, whose data is processed by the YOLOv11-Pose deep learning model for human detection and hand signals, to provide visual perception. The triangulation principle based on the pixel distance between the head and neck keypoints is utilized to calculate the distance, during which the hand gesture interpretation system sets the robot operational status. The results indicate that the F1-score of the object detection model is greater than 0.92, with an inference speed of 14 FPS. While the human distance estimation algorithm recorded high accuracy, it achieved an average Mean Absolute Percentage Error (MAPE) of below 5.3%. The reliable hand gesture detection system is proven by average detection confidence values above 0.83 at distances of 1.5 to 5 meters. On the supermarket simulation track, the robot responded and followed the human movement stably and successfully. Moreover, it can perform several commands using hand signals for huma. It follows stop mode activation and moves forward, backward, and maneuvering.

1. Introduction

Either online or in-person shopping plays a crucial role in everyday life. Although online shopping gains its popularity, in-person shopping experience offered by physical retail stores such as supermarkets have been maintained as it allows consumers to physically evaluate and interact with the products, as shown by Thompson *et al.,* [1] and Wang *et al.,* [2]. The surveys reveal that supermarkets are the most preferred to buy daily needs.

However, challenges dealing with the supermarket shopping experience remain. The challenges reduce both efficiency and convenience. Moreover, besides overcrowding, it is difficult to find products. These dynamic environments pose significant navigation and safety risks for autonomous

* *Corresponding author.*
E-mail address: wahyuditinom@elektro.undip.ac.id

systems, while the physical burden of pushing heavy trolleys remains a primary concern for customers, as reported by Purwantono *et al.,* [3] and Vikash Ranjan and Akhtar, [4]. One of the most crucial contributing factors to customer satisfaction, as discussed by Vikash Ranjan and Akhtar, [4] is service quality. It includes physical shopping experience.

A human-following robot serves as an innovative solution to address the physical shopping challenges, as reported by Purwantono *et al.,* [3]. These systems aim to remove the physical burden of pushing trolleys, which in turn promotes efficiency, as demonstrated by Ramzan *et al.,* [5]. However, there are two main challenges in target reidentification (Re-ID) to implement the systems. The first paradigm, appearance-based tracking, uses visual features such as clothing color or deep learning features to recognize targets, as shown by Tsai and Yao, [6]. Although nonintrusive, this method is highly susceptible to real-world conditions. Tracking performance can degrade significantly when the target is occluded or others with similar colored clothing appear.

On the other hand, the second paradigm that uses visual markers such as AruCo and AprilTag offers highly reliable identification, as reported by several authors [7–9], but at the expense of user experience. The users are burdened with actively and constantly pointing out the marker to the robot. This interaction is unnatural and prone to complete failure if the marker is covered even a fraction.

Accurate distance estimation is crucial in human-following systems, maintaining a safe distance and smooth movement. To achieve high precision, the use of specialized sensors such as LiDAR and depth cameras is the first choice, as reported by several authors [10–13]. Studies show that these sensors are capable of producing very low Mean Absolute Errors (MAE); LiDAR sensors can achieve MAEs as low as 3.27 to 4.91 cm within their operational range, as reported by Reddy Kavya Sree *et al.,* [10] and Pinnamaraju *et al.,* [11], While depth cameras such as Intel RealSense show an average position error on the distance axis (Z-axis) of about 6.57 cm, as shown by Suwandi *et al.,* [12] and Mingozzi *et al.,* [13]. Although highly accurate, the implementation of these sensors is often hampered by significant hardware costs, which limits large-scale commercial scalability and adoption in the retail industry. As a more economical alternative, vision-based approaches using a single RGB camera with triangulation methods have become popular as they can drastically cut costs. Geometric projection becomes the basis of this method. To estimate the distance, geometric projection based on the width of the object detection result's bounding box and the target's assumed shoulder width is required, as discussed by Herdianto *et al.,* [14] and Wicaksana *et al.,* [15]. Although this is considered cost-effective, the effectiveness strongly depends on the target's orientation towards the camera. The weakness of this method deals with pose changes, if the body is not facing straight to the camera, the error of shoulder width-based distance estimation method can be more than 12%, as reported by Wicaksana *et al.,* [15]. When the target rotates, the shoulder width projected on the 2D image will narrow significantly. The distance calculation becomes invalid. Therefore, this model should be compensated with a more complex pose model.

While advances in tracking and distance estimation continue to develop, the following fundamental challenge lies in the often-overlooked aspect of human-robot interaction (HRI). Existing human-following robot systems often operate in an 'always-on' mode or require cumbersome control interfaces, such as using a Graphical User Interface (GUI) or physical buttons on the robot, as reported by Zachariae *et al.,* [9] and Alhmiedat *et al.,* [16]. These methods break the user's natural interaction flow, forcing them to shift their focus from shopping to give simple commands such as 'stop' or 'continue'. This lack of intuitive, contactless interfaces is a significant barrier to user acceptance and technology adoption in dynamic public environments. Therefore, a crucial research gap opens up to develop a control system that integrates seamlessly into the robot's perception flow, allowing users to give commands naturally without disrupting their shopping movement.

Based on this analysis, the research gap is not in the lack of solutions to individual problems, but rather in the lack of a systemic architecture that can integrate solutions to these conflicting trade-offs. There is a need for a human-following service robot system that simultaneously combines an unambiguous and cooperative target acquisition mechanism, a computationally efficient unified multi-task perception flow for edge devices, and a seamlessly integrated contactless control interface, using only a low-cost monocular RGB camera sensor.

An integrated supermarket robot prototype is proposed, offering systemic integration of a Unified Perception Engine on a resource-constrained edge device using only a low-cost monocular camera. This makes the proposed prototype different from existing human-following and gesture-control systems. Moreover, the proposed architecture presents three distinct systemic novelties, distinguishing it from those that use separate, computationally heavy deep learning pipelines for tracking and gesture recognition, or rely on active markers that demand constant user attention. The first distinct systemic novelty is a Hybrid Tracking Protocol. Different from the existing marker-based systems that require the user to constantly point the tag at the robot, the proposed system only uses AprilTags as a user-initiated opt-in mechanism for unambiguous initial target acquisition. Once locked, it switches to a Kalman Filter-enhanced visual tracking, allowing natural movement, as demonstrated by Zhang *et al.,* [17]. The second distinct systemic novelty is a Unified Perception Engine. It is an engine used to overcome the computational bottlenecks of running multiple models. A single YOLOv11-Pose model was custom-trained to perform three complex tasks simultaneously in one inference run: human detection, keypoint extraction, and hand gesture recognition, as reported by Rasheed *et al.,* [18] and Liu *et al.,* [19]. The third distinct systemic novelty, an Orientation-Invariant Triangulation method, used for distance estimation. The conventional monocular methods rely on shoulder-width that fails because of maneuvers. Meanwhile, this method ensures stable tracking during dynamic retail maneuvers by using the vertical distance between head and neck keypoints, as shown by Wicaksana *et al.,* [15] and Petković *et al.,* [20]. Finally, hand gesture control is seamlessly integrated into the same unified perception engine, creating a highly efficient and intuitive perception-to-control loop, as discussed by Herbaz *et al.,* [21] and Zhou *et al.,* [22]. Through this systemic integration, this research aims to demonstrate a reliable, efficient, and user-friendly human-following robot, specifically designed for dynamic retail environments.

2. Materials and Methods

To functionally create the robot prototype, hardware and software were integrated. NVIDIA Jetson Orin Nano was the central brain of the robot. It handled vision processing and deep learning inference, as demonstrated by Liu et al*.,* [23] and Islam et al*.,* [24] and offered balanced AI inference performance and power efficiency. 32 Tensor Cores were equipped with 1024-core Ampere architecture GPU, providing the necessary high-level computing power while remaining suitable for mobile robotic applications. The configuration of hardware system is shown in Figure 1.

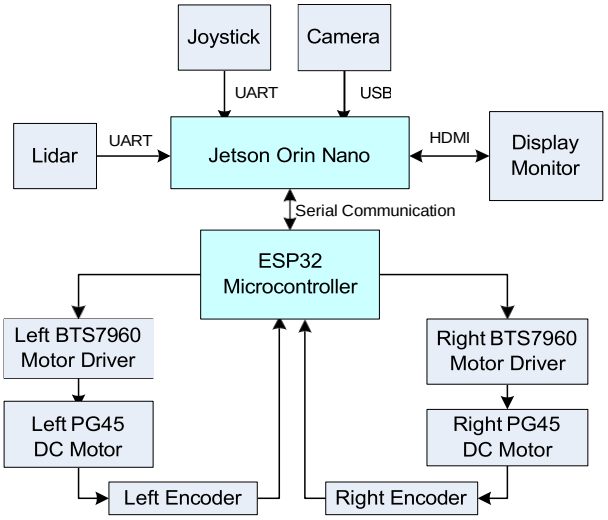


Fig. 1. Hardware design for a human following system

The design covers a monocular RGB camera (M-Tech WB500). This USB-connected camera acts as the primary perception sensor. The actuation involves Jetson Orin Nano that communicates with an ESP32 microcontroller serially. Besides, it helps translate linear and angular velocity commands into PWM signals. The PWM signals are responsible to control two DC motors. This component is useful to handle time-sensitive operations. It generates precise Pulse-Width Modulation (PWM) signals for the motor drivers and processes sensor data, freeing the Jetson Orin Nano from such critical tasks. Communication between the Jetson and ESP32 is established via a UART serial link, through which high-level velocity commands (linear and angular) are transmitted.

2.1 Yolov11-Pose Custom Model Design

To perform human, keypoint estimation, and hand gesture recognition simultaneously; a custom deep learning model based on the YOLOv11Pose architecture is developed. The model's performance relies on a comprehensive dataset curated specifically for this task. For the dataset, there are a total of 6,650 images. The combination of the public data with custom images, captured directly from the robot's camera, accurately represent the operational environment. The four classes of this dataset are: person, stone, paper, and scissors. The are a total of 10,299 annotations for training data, 2,846 for validation data, and 1,439 for test data. Roboflow platform performed the annotation process. To localize the objects for all classes, bounding boxes were drawn. For the person class, two important keypoints were also annotated: keypoint 0 (k0) at the top of the head and keypoint 1 (k1) at the center of the neck. An example of image annotation using Roboflow is shown in Figure 2.

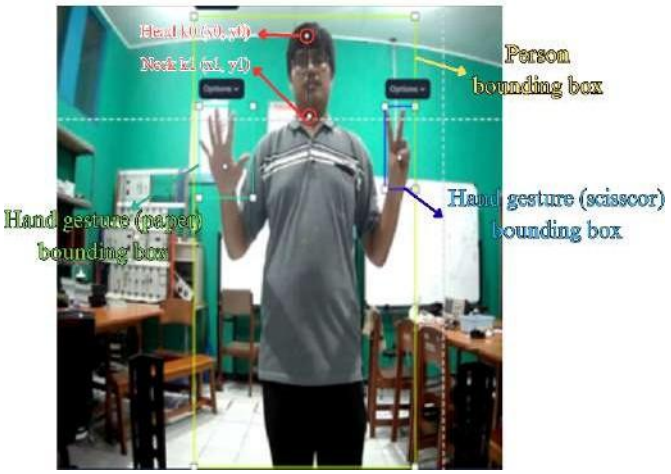


Fig. 2. Example of image annotation using Roboflow

To train a YOLOv11-Pose, a transfer learning approach was used. It is an approach that initializes the network incorporating pre-trained weights to speed up convergence. NVIDIA T4 GPUs, through Google Collaboratory platform, was utilized in the training. The main framework is PyTorch. Key hyperparameters for the training process include an input image size of 640x640 pixels and a training duration of 150 epochs. Once the training is successful, the file of PyTorch model (.pt) undergoes optimization step, converted into the TensorRT engine directly on the Jetson Orin Nano. Through this conversion, the computational graph is optimal as the layers are merged and the hardware-specific-kernels are selected. While retaining accuracy, FP32 precision reduces inference latency sharply and maximizes throughput on the Jetson hardware, as reported by Li *et al.,* [27].

2.2 Human-Following System Design

The design, human-following system, makes it possible for the robot to track and follow the users dynamically, with 5-meter effective estimated distance limitation. Through the Hybrid Tracking, one unified flowchart shows the detection as well as the tracking workflow. SEARCHING mode is the initial process. On each image frame, the YOLOv11-Pose model and AprilTag detector run simultaneously. Once an individual associated with AprilTag ID 0, target acquisition occurs, the individual is locked as the primary target. The system then switches to TRACKING mode. This initial state initializes the Kalman Filter algorithm, through which linear system states using new measurements are predicted and corrected. This mechanism effectively tracks movement while maintaining target identity, even during momentary visual occlusions, as reported by Pan *et al.,* [28], and Kumar *et al.,* [29]. Once the target is locked, the system no longer relies on AprilTag. The Kalman Filter predicts the target position for each subsequent frame based on the motion model, matching all human detections from YOLOv11-Pose in the current frame. One with the highest match score (based on positional proximity) will be re-associated as the target, ensuring continuous tracking even when AprilTag or the user's entire body is temporarily obstructed.

The concept of triangulation performs the distance estimation once the target is identified. It will be highly susceptible to changes in user pose if the conventional distance estimation methods based on bounding box width or shoulder width are used. The error will be more than 12% when the target is not facing straight into the camera, as shown by Herdianto *et al.,* [14]. Therefore, the vertical pixel distance between k0 and k1 is used as it is more invariant to body rotation (yaw), providing more stable and accurate distance estimation in dynamic movement scenarios. Camera calibration is

crucial before distance estimation is camera calibration, used to determine its intrinsic parameters. This process uses the camera_calibration package from ROS with the checkerboard method, as reported by Schramm *et al.,* [30]. The camera intrinsic matrix (K) is obtained from this process, as shown in Eq. (1).

$$K = [fx\ s\ Cx\ 0\ fy\ Cy\ 0\ 0\ 1\ ] = [562{,}77\ 0\ 329{,}56\ 0\ 561{,}22\ 237{,}35\ 0\ 0\ 1\ ] \quad (1)$$

From this matrix, the focal length values in pixel units are obtained: fx =562.77 and fy =561.22. Since our distance estimation algorithm is based on the vertical dimension, the value fy is used as the key parameter in the subsequent calculations.

The k0 and k1 coordinates of the target are extracted. The vertical distance between these two points in pixels (P) is calculated using the Euclidean distance equation shown in Eq. (2), as reported by Shih *et al.,* [31] and Chen *et al.,* [32].

$$P = 5\overline{(x_0 - x_1)^2 + (y_0 - y_1)} \quad (2)$$

Knowing the absolute reference height between the head and neck (H, assumed to be 25 cm) and the camera focal length (F, i.e., fy from calibration), the distance (D) to the target is estimated using the triangular bisection principle of the pinhole camera model, as formulated in Eq. (3), as reported by Petković *et al.,* [20] and Duarte *et al.,* [33]. The triangular bisection principle is shown in Figure 3.

$$D = \frac{F \times H}{P} \quad (3)$$

This distance information, along with the horizontal position of the target (x coordinate of the head keypoint), is published to the ROS topic /yolo/person_info.

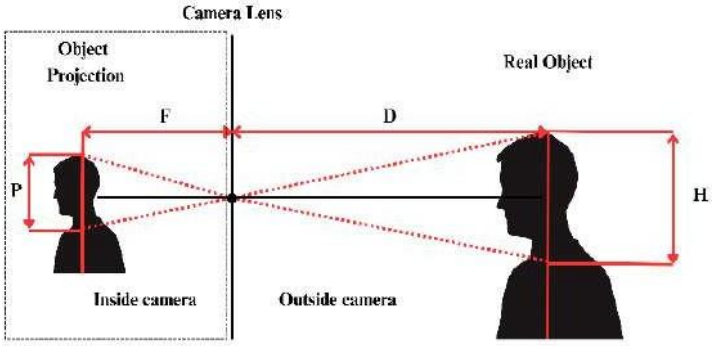


Fig. 3. Triangular bisection principle

To receive distance and position data, a control node subscribes to the topic /yolo/person_info. This data first passes through a simple filter and a timeout mechanism (1 second) to handle fluctuations and target loss. There are three distance zones to determine the robot's linear velocity. If the distance exceeds 1.85 m, it moves backward. Meanwhile, if the distance is below 1.4 m, it moves forward. The intermediate-range distance makes the robot remains stationary. The straight of three robot zones is shown in Figure 4 and the sidewise of three robot zones is shown in Figure 5.

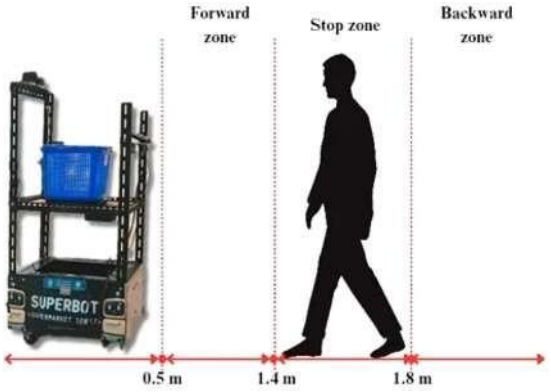


Fig. 4. Straight of three robot zones: forward zone, stop zone, and backward zone

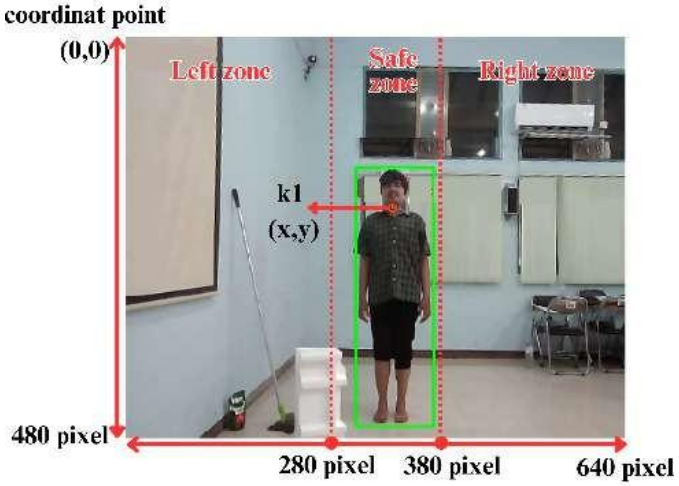


Fig. 5. Sidewise of three robot zones: left zone, safe zone, and right zone

The angular velocity is calculated proportionally to the target's horizontal distance from the Safe Zone boundary, which is defined between pixels 300 and 380 in this code. The benchmark for this calculation is the k1 coordinate value obtained from the k1 keypoint on the detected target. This logic allows the robot not to make corrections when the target is already inside this center area, resulting in more stable and efficient movements and avoiding unnecessary oscillations.

The target velocity value is applied through ramping mechanism, acceleration or deceleration mechanism. It is not directly applied to produce smooth motion. Following the process, the value of the final velocity is then published to the motor actuators. Figure 6 shows the algorithm flow for the human-following.

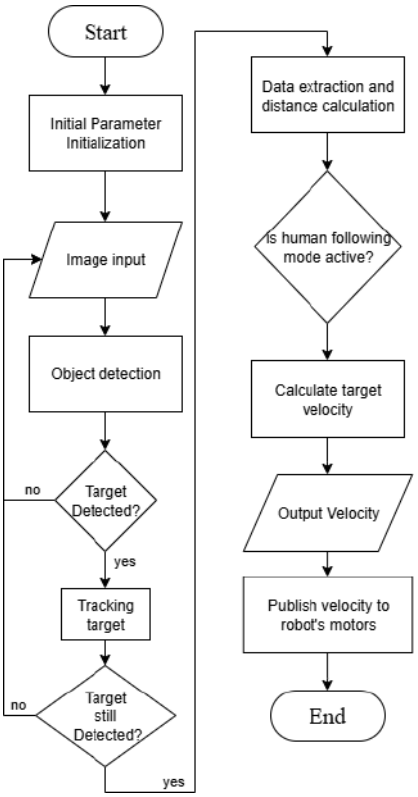


Fig. 6. Algorithm flow for the human following system

2.3. Design of Hand Sign Interpretation System

A contactless and intuitive control interface are parts of the system, a system that focuses on three specific hand signals. The hand signals were adopted from the rock-paper-scissors game. ‘Rock' signals STOP, 'paper' activates HUMAN_FOLLOW mode, and 'scissors' signals GO. Figure 7 shows the workflow of this system. This system begins with image acquisition from the camera. Each frame is analyzed by the YOLOv11-Pose model to detect the presence of hand gestures. The system applies a strict validation logic, ensuring that commands come from authorized users only.

Only the gestures from the authorized user are processed. This requires initial identity validation. If its bounding box falls within the primary target’s (ID 0) detection area, established during the AprilTag acquisition phase, a gesture is considered valid. Gestures, from non-authorized users, will be ignored. The second performed validation is ambiguity validation. The condition is invalid if, in a single frame, two or more hand signals are detected simultaneously. To prevent the incorrect command interpretation, the detection histories are all reset. The detection of a valid gesture leads the system to debouncing, entering the time-based confirmation. An at least 1.0 second consistent detection is required to filter momentary detection. Before the signal is mapped to a robot command, this debouncing stage needs 9 consecutive frames. Through this mechanism, momentary detections, accidental gestures, are filtered out. The corresponding string command ('STOP', 'GO', or 'HUMAN_FOLLOW') maps the validated gesture, that later published to the ROS topic /yolo/gesture_detected. A controller state manager node will subscribe to this topic and change the robot's operating mode dynamically, ensuring a smooth and structured transition between behavior modes. The algorithm flow of the hand gesture interpretation system is presented in Figure 7.

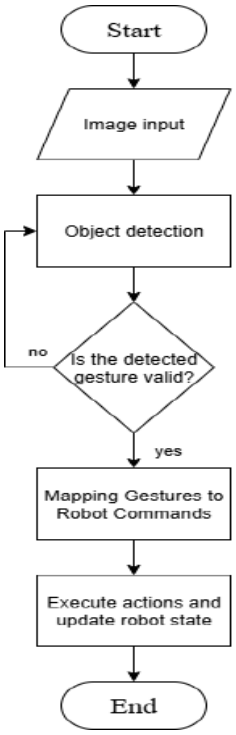


Fig. 7. Algorithm flow for interpreting hand gesture

3. Results and Discussion

Each functional component is validated through a comprehensive system testing. The validation covers the performance of the deep learning model to the overall behavior of the robot in a simulated environment.

3.1 YOLOv11-Pose Model Performance Evaluation

The purpose of this test is to ensure the validation of the YOLOv11-Pose custom model performance and effectiveness. This ensures that the model can detect human poses and hand gestures. There are several aspects in the evaluation process. The loss value during training is measured and the classification performance is tested. The standard evaluation metrics help measure its accuracy, precision, recall, and F1-score. The model should be able to distinguish different classes. Thus, A confusion matrix is used. In addition to classification accuracy, inference speed was tested, primarily by measuring the Frames Per Second (FPS) value and average inference time after the model was converted into the TensorRT engine format.

To test the effectiveness of model training, a graphical analysis of loss values reflecting the learning process of the model during training was conducted. The four metrics analyzed were train/box_loss, train/cls_loss, val/box_loss, and val/cls_loss. The train loss metric reflects model error on the training data, while val loss indicates how well the model can generalize to new data. Box loss describes how correctly the location and size of the bounding box are predicted, while class loss shows the model's classification ability to label correctly. Loss graphs can also help identify problems such as underfitting or overfitting. With train/box_loss around 0.75 and val/cls_loss around 0.39, all loss metrics are comparatively high at the 25th epoch. An active learning phase occurs as there is a

sharp decrease in all losses between epochs 25 and 75. The decrease continues at epochs 75-125, albeit slower, and begins to level off near optimal performance. In the final phase (epochs 125-150), the decrease remains gradual, with val/box_loss reaching 0.56 and val/cls_loss 0.24 at the end of training. Overall, the loss graph shows a stable convergence process with no sign of overfitting. The model performance analysis shows initial convergence at the 25th epoch with an mAP50 of 0.97, although the localization precision (mAP50-95: 0.741) still requires optimization. Until the 100th epoch, there was a substantial increase in mAP50-95 to 0.789, while mAP50 peaked at 0.98. In the final phase (epochs 100-150), mAP50 saturated, but mAP50-95 continued to increase to 0.793. This trend indicates that once the basic detection capability is achieved, training effectively optimizes the accuracy of the bounding box. Thus, the optimal model performance that balances detection and localization is achieved at the 150th epoch.

Classification performance was analyzed using the confusion matrix on the test dataset. From the confusion matrix, the True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values for each class were calculated. The total predicted data analyzed through the confusion matrix is 2959, consisting of 2846 test data and 113 non-classified data. From this total, each class's performance calculations were carried out using advanced classification evaluation metrics, including accuracy, precision, recall, and F1-score. This calculation is based on the TP, FP, FN, and TN values, using Eq. (4) to (6).

$$Precision = \frac{TP}{TP)FP} \tag{4}$$

$$Recall = \frac{TP}{TP)FN} \tag{5}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision)Recall} \tag{6}$$

The results show a powerful performance in most classes. The 'person' and 'rock' classes achieved an F1-score of 0.98 and 0.97, respectively, indicating reliable detection. The 'scissors' class also showed solid performance with an F1-score of 0.95. The model's main challenge lay with the 'paper' class, which recorded a lower precision (0.87) despite a high recall (0.98), resulting in an F1 score of 0.92. This suggests that the model sometimes misclassified other cues as 'paper', possibly because the silhouette of the open hand bears resemblance to background objects or other cues; however, with all classes achieving an F1-Score above 0.92 and a global model accuracy of 0.95. Model classification is shown in Table 1. Speed testing was performed after converting the model to TensorRT engine format and running on an NVIDIA Jetson Orin Nano. The results were very satisfactory, with the system achieving an average inference speed of 14 FPS and an average pure model inference time of 21 ms. This performance quantitatively validates the architectural advantages of the Unified Perception Engine. The system can operate in real-time on resource-constrained edge devices by executing three perception tasks in a single model. This achievement is difficult to achieve with a separate multi-model approach.

Table 1
Model classification succes rate based on the confusion matrix

Class	Precission	Recall	F1-Score
Stone	0.95	0.98	0.97
Scissor	0.93	0.97	0.95
Paper	0.87	0.98	0.92
Person	0.98	0.99	0.98

The model proved effective for the designed task with an F1-score above 0.92 for all classes and a global accuracy of 0.95. The performance is comparable to the report on Deep Learning Empowered Hand Gesture Recognition: using YOLO Techniques by Herbaz *et al.,* [21], which achieved accuracy above 97% for single hand gesture recognition. However, in contrast to those single approaches, the YOLOv11-Pose model in this study is designed as a multi-task architecture that simultaneously performs human detection, keypoint pose extraction, and hand gesture recognition in a single inference. The main advantage of this approach lies in computational efficiency. With a speed of 14 FPS on Jetson Orin Nano, this system surpasses the performance reported in Equipped with Monocular Depth Estimation and Intelligent Wake-up Vision Based Tracking System for a Human-Following Mobile Robot by Tsai and Lee, [34], which only achieves 5.57 FPS on Jetson AGX Xavier. Moreover, our performance is comparable to that of Human Tracking and Following using Machine Vision on a Mobile Service Robot by Yong *et al.,* [35], which achieves 15 FPS. These results confirm the effectiveness of the proposed Unified Perception Engine strategy, which can handle three complex perception tasks without sacrificing operational speed on edge devices.

3.2 Confidence Value of Hand Sign Detection

Confidence value testing was conducted to assess the system's ability to detect hand gestures from a distance of 150 to 500 cm, with two variations of hand orientation: facing and facing away from the camera. Focusing on the stone gesture class, the system showed a relatively stable detection performance. In the hand position facing the camera, the highest confidence value of 0.85 was recorded at 150 cm and decreased slowly to 0.78 at 500 cm, remaining above 0.80 until 400 cm. The confidence value ranged from 0.79 to 0.86 at the back-to-camera position, with a peak at 200-250 cm. The average confidence of both variations is 0.83 and 0.84, indicating that the system can detect the stone cue consistently despite the difference in hand orientation. The full values are in Table 2.

Table 2
Test results of detection confidence values for the ‘rock’ class in two variations

Actual	Confidence Value (0-1)	
Distance (cm)	Variation 1	Variation 2
150	0.85	0.85
200	0.85	0.86
250	0.84	0.86
300	0.84	0.84
350	0.83	0.84
400	0.85	0.84
450	0.82	0.79
500	0.78	0.80
Average	0.83	0.84

In the scissor gesture class test, the system showed good detection consistency at various distances and hand orientations. In the first variation, confidence values ranged from 0.82 to 0.86, with the highest value at 150 cm. In the second variation, the highest confidence value was recorded at 0.87 at 150 cm, and the lowest at 0.78 at 500 cm. The average confidence of both variations was 0.84, indicating the stability of the gesture detection despite the change in hand orientation. A significant decrease was only seen after 450 cm, possibly due to shrinking the bounding box or losing hand shape detail. The complete data is in Table 3.

Table 3
Detection confidence values for the 'scissors' class in two variations

Actual	Confidence Value (0-1)	
Distance (cm)	Variation 1	Variation 2
150	0.86	0.87
200	0.84	0.86
250	0.85	0.86
300	0.86	0.86
350	0.85	0.85
400	0.86	0.81
450	0.81	0.79
500	0.81	0.78
Average	0.84	0.84

In the paper hand gesture class test, the system showed consistently high detection confidence values for both hand orientation variations. For the first variation, the highest confidence value reached 0.91. In the second variation, the confidence values ranged from 0.83 to 0.90. Thus, 0.87 and 0.85 are the average confidence values of these two variations. This shows that the system, over a wide range of distances, can generally and stably detect paper gestures. The more open shape of the paper hand may lead to this good performance as visual features are more easily recognized. Table 4 presents the confidence value data for the paper class.

Table 4
Detection confidence values for the 'paper' class in two variations

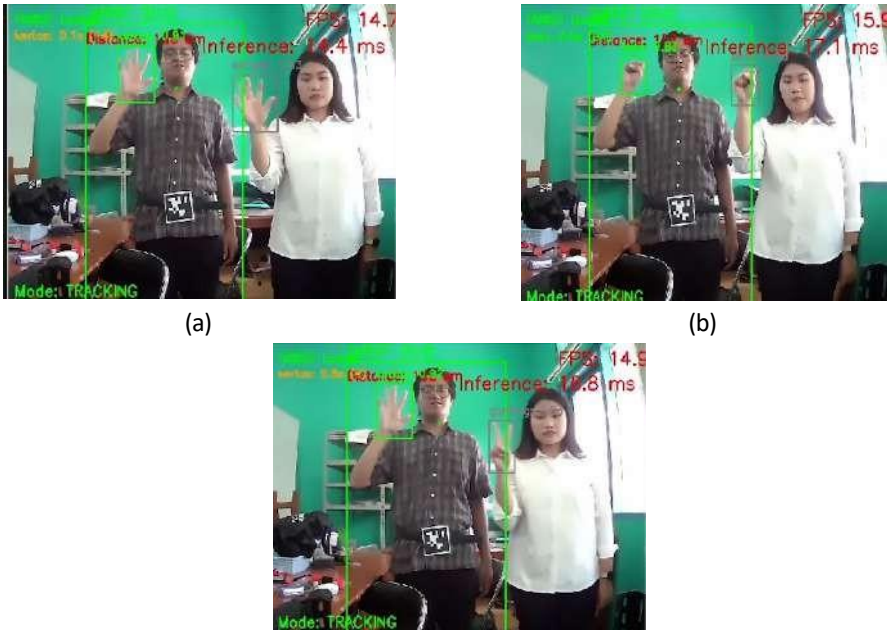
Actual	Confidence Value (0-1)	
Distance (cm)	Variation 1	Variation 2
150	0.91	0.88
200	0.91	0.85
250	0.87	0.90
300	0.88	0.85
350	0.85	0.85
400	0.84	0.83
450	0.83	0.82
500	0.83	0.83
Average	0.87	0.85

The test results of the hand gesture interpretation system can be compared with Long-Range Hand Gesture Recognition via Attention-based SSD Network by Zhou *et al.,* [22] which targets explicitly long-range hand gesture recognition with a range of up to 7 meters. However, a complex network architecture, focusing on one task, are necessary to achieve these. In contrast, the system developed in this study can achieve an interaction range of 5 meters relevant for the supermarket

context, while handling human detection and pose estimation in a single model. The added advantage of the 2-meter range extension is not worth the increased complexity and computational requirements on an edge-based robotic platform. Thus, this study demonstrates that a well-trained, versatile detector can serve as an efficient and integrated HRI module, offering a more practical approach than developing specialized models for each perception sub-task.

3.3 Validity of Hand Signals

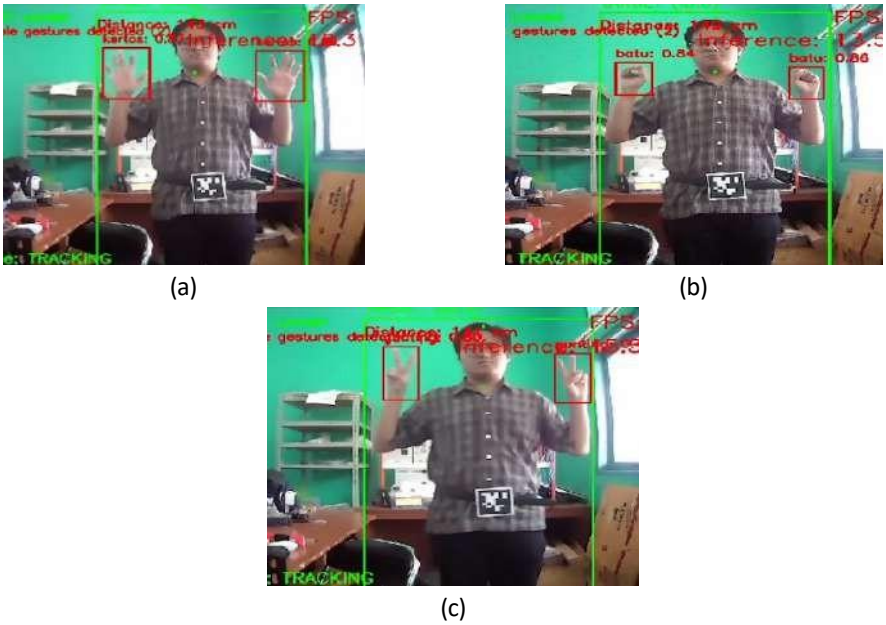
The hand gesture-based robot control system was tested to evaluate the system's accuracy in processing and transmitting commands. The testing was divided into two main stages. The first is gesture validity testing, which ensures the system only responds to gestures from users with ID 0 and will ignore commands if more than one gesture is detected in a single frame. The second stage is command delivery testing, where the rock, scissors, and paper cues are associated with the STOP, GO, and HUMAN_FOLLOW commands, respectively. A command will only be sent if a valid cue is detected continuously for at least 1 second. Hand gesture detection is shown in Figure 8. Based on Figure 8(a), the paper gesture from the person with ID 0 was successfully detected and validated as a command, indicated by the green bounding box on the user's hand. On the other hand, similar gestures from other people are also detected, but ignored by the system as they are not from the active ID, indicated by the gray bounding box. In Figure 8(b), the stone gesture from ID 0 is also recognized and accepted as a command, while the gestures from others are ignored. Figure 8(c) shows that the scissors cue from ID 0 is again validated as a command, while those from others remain ignored. Thus, the system can selectively distinguish and validate hand gestures based on the active user ID. Next, the validity of the number of gestures was tested with a scenario with more than one hand gesture in the frame. The results of the second validity test are shown in Figure 9.



(a) (b)

(c)

Fig. 9. Hand gesture detection: green boxes for valid detections and gray boxes for invalid detections for the gestures (a) paper, (b) rock, and (c) scissors



(a) (b)

(c)

Fig. 10. Invalid gesture detection due to more than one gesture being detected, indicated by red boxes for gestures (a) paper, (b) rock, and (c) scissors

Based on Figure 9(a), two paper cues are detected, but both are considered invalid as the system is designed to accept only one cue at a time. Similarly, the system rejects all such multiple inputs when two stone cues are detected in Figure 9(b) and two scissors cues in Figure 9(c). In each case, the invalidity of these commands is indicated by the presence of a red bounding box around both hands.

3.4 Distance Estimation Accuracy

The vision system results were compared with the meter measurements to ensure the distance estimation accuracy. This process involved MAPE as an error indicator, The average MAPE of 3.69% shows that the accuracy is high in the variation of targets facing the camera. If the object was partially cut off or too close, slightly larger errors appeared at close distances (5.33% at 150 cm and 6.50% at 200 cm). At medium to long distances, however, accuracy improves. The errors are below 5%, reaching 1.00% at 500 cm. Thus, the performance system is reliable and stable. Table 5 shows the test data on the back-to-camera variation.

Table 5
Test results of the distance estimation algorithm with the subject facing the camera

Actual Distance (cm)	Estimated Distance (cm)	Mean Absolute Percentage Error (%)
150	158	5.33
200	213	6.50
250	258	3.20
300	292	2.67
350	367	4.86
400	385	3.75
450	460	2.22
500	505	1.00
Average Error (%)		3.69

The second variation was performed when the target had its back to the camera. In this variation, significant performance anomalies were found. At close distances (150 cm and 200 cm), the algorithm produced high errors of 18.67% and 12.50%. However, at distances of 250 cm and above, the accuracy improved dramatically with very low errors, even reaching 0.57% at a distance of 350 cm. At close distance, however, the value is strongly influenced by inaccuracies, although the overall average MAPE is 5.27%. Partial occlusion at the neck keypoint (k1), such as by hair, shirt collar, or a head hood causes error anomalies, particularly at a distance of 150 cm and 200 cm when the target has its back to the camera. Vertical pixel distance between the head and neck is the core of this method. Thus, inaccurate distance estimation may occur at any point, especially when the pixel projection is large at close distances. However, the target projection size becomes smaller and more stable within the camera frame once the distance increases. This offers more consistent keypoint detection and drastically improves accuracy. The test data on the back-to-camera variation is shown in Table 6.

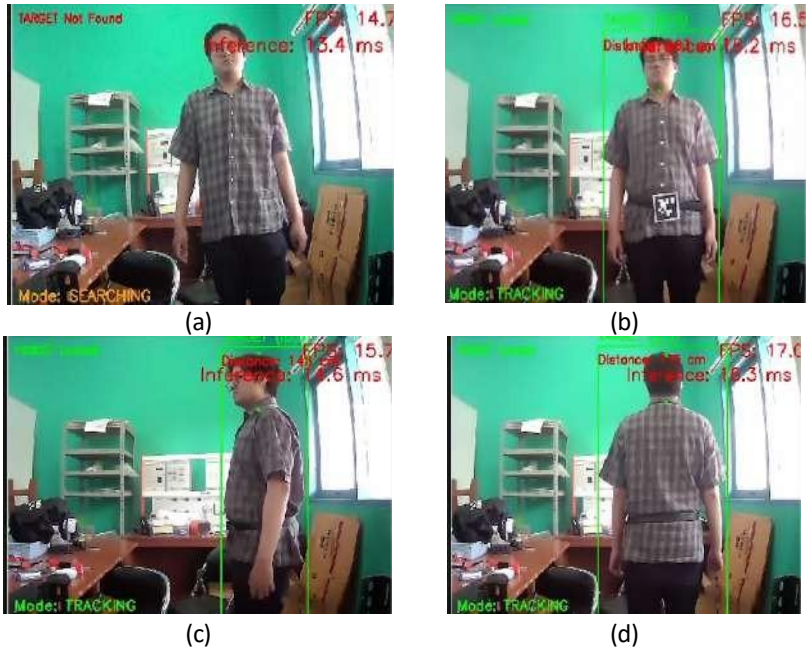
The distance estimation results in this study show competitive accuracy compared to Shoulder Keypoint-based Distance Estimation for Human-Robot Interaction by Wicaksana *et al.,* [15], which reports an average error of 1.3% over a limited distance range of 60-180 cm under ideal conditions. In contrast, the proposed head-neck vertical distance-based approach maintains an average MAPE below 5.3% over a wider operational range of up to 5 meters, even when the target has its back to the camera. This confirms the robustness and flexibility of the method to dynamic pose variations in real environments.

Table 6
Test results of the distance estimation algorithm with the subject facing away from the camera

Actual Distance (cm)	Estimated Distance (cm)	Mean Absolute Percentage Error (%)
150	178	18.67
200	225	12.5
250	255	2.00
300	292	2.67
350	352	0.57
400	393	1.75
450	461	2.44
500	492	1.6
Average Error (%)		5.27

3.5 Person identity tracking capability

Person identity tracking capability tests were conducted to continuously evaluate the system's consistency in maintaining detected human objects' identity (ID). The design is the referent of this test. AprilTag visual marker, used for initial identity acquisition, and the Kalman Filter algorithm, used for continuous tracking, are combined in the system. The order of appearance is no longer used to determine the identity of the primary target. It is the person carrying the AprilTag with a specific ID (ID 0) is detected. This target acquisition process occurs while the system is in SEARCHING mode. Once the target is successfully locked, the system switches to TRACKING mode, where the Kalman Filter takes over to predict and maintain the target's trail, even when the AprilTag is no longer visible to the camera. An illustration of the results of this test is shown in Figure 10.



(a) (b) (c) (d)

Fig. 10. Object tracking test results: (a) Initial system state in SEARCHING mode, (b) Target acquisition moment when AprilTag ID 0 is detected, (c) and (d) Successful re-identification when the AprilTag is occluded

The test in Figure 10 shows an individual tracking system, with only one target with AprilTag ID 0. In Figure 10(a), the system is still in SEARCHING mode because the AprilTag is not yet visible. Once the tag is detected (Figure 10(b)), the system switches to TRACKING mode and successfully locks onto the target while ignoring other individuals. Figure 10 (c-d) tests the reidentification when the tag is not visible. The system still tracks the target accurately thanks to the Kalman Filter prediction, proving the tracking robustness even if the tag is lost from sight.

The dilemma in human-following systems is tackled with the proposed Hybrid Tracking Protocol. In mixed environments, the Human-Following Strategy for Mobile Robots shows the benefit of LiDAR-based foot tracking. It is resistant to visual occlusion and independent of appearance. However, lacks an explicit identity locking mechanism. Thus, when individuals are cross or close to each other, the system may lose its target consistency. On the other hand, Color Histogram-based Human Following System [6], as one of pure appearance-based approaches, is liable to failure. The presence of individuals wearing similar clothing colors causes severe. Through this developed system, the limitations are addressed, combining AprilTag and visual detection. AprilTag handles opt-in initialization. The user does not have to constantly display the marker to guarantee unambiguous identification. Thus, the system combines the reliability of tag-based identification with the flexibility of visual tracking, resulting in a more robust and intuitive user experience.

3.6 Robot Motion Response in Following a Human on a Supermarket Track

The ability of the robot to stably follow human movement is tested. Tests were carried out in three variations in Room B206, Electrical Engineering, Diponegoro University, using a simulation of shopping activities in a supermarket, where the robot followed the user who moved around the location in the test area with a coordinated map. The results of the test for the first variation are shown in Figure 11.

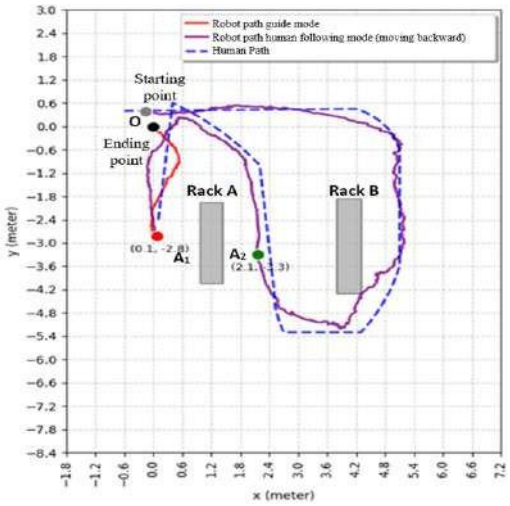


Fig. 11. Path comparison between human movement and robot movement on the A1A2O route

System testing began with the establishment of waypoints A1 and A2 via the GUI, where a 'scissors' hand gesture successfully initiated the guiding mode for autonomous navigation towards A1, with mode transitions validated by status on the interface. Upon reaching the proximity of A1 at coordinates (0.1, -2.9), the system automatically completed the sub-task, then switched to human following mode via the 'paper' gesture, which the GUI again confirmed. In this mode, the robot executes backward motion while maintaining distance and performs re-centering maneuvers based on the user's lateral position to maintain alignment. During the dynamic maneuvers, the cue detection is reliable as the 'rock' cue precisely stopped the robot on reaching A2 at (2.1, 3.3). The robot followed human operations until endpoint O after task 2 was manually completed on the GUI. This indicates that the system is capable to integrate goal-based navigation, cue control, and reactive maneuvering coherently.

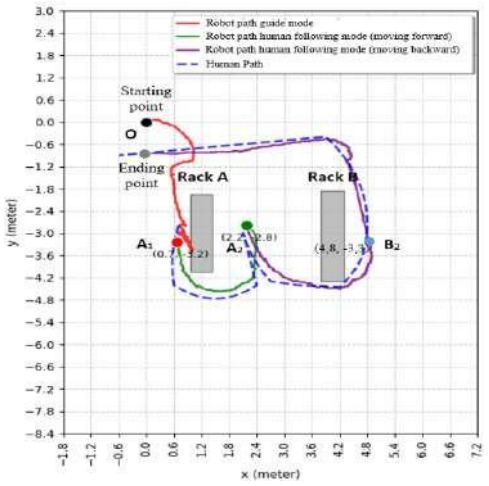


Fig. 12. Path comparison between human movement and robot movement on the A1A2B2O route

The analysis of the robot motion response on variation 2, i.e., the A1A2B2O supermarket trajectory, whose path comparison is illustrated in Figure 12, comprehensively validates the robustness and flexibility of the multimodal interaction system design. The system successfully

integrated a GUI interface for strategic planning (determination of waypoints) with hand gesture recognition for tactical control in real-time, allowing seamless transition between guiding mode, forward human following mode, and backward human following mode. The reliability of this design is reflected in the system's ability to precisely interpret the stop ('stone') command while the robot is in motion. Furthermore, analysis at point B2 revealed a crucial implicit safety behavior: the robot stopped automatically without an explicit command when the user was at a very close distance (<1.3 m), caused by a failure of full-body keypoint detection. Although this phenomenon is a sensory limitation, it functionally acts as a passive safety mechanism that prevents unexpected movements when tracking data is no longer reliable. Overall, the high correlation between robot and user trajectories has validated the robot's capability to switch between complex motion modes while maintaining stable tracking of the user's path.

Motion response analysis on the A1B1B2O route, whose path comparison is illustrated in Figure 13, validated the effectiveness and flexibility of the hybrid control architecture in a multi-stage scenario. This test effectively demonstrated the integration between the initial autonomous navigation (guiding mode) activated by the 'scissors' cue, with the forward continuous visual tracking (human following) initiated by the 'paper' cue after the first target achievement automatically. The steering algorithm design proved robust, with the robot able to perform precise re-centering maneuvers based on the user's lateral position to maintain alignment during forward movement. Furthermore, the system demonstrated its ability to handle a series of discrete commands sequentially, responding accurately to two stop commands ('stone' cues) at points B1 and B2, which were then combined with manual task completion via GUI interaction. The workflow combination of real-time cue detection for motion control and GUI interaction for task management confirms the system's robust and adaptive design for complex, step-by-step navigation.

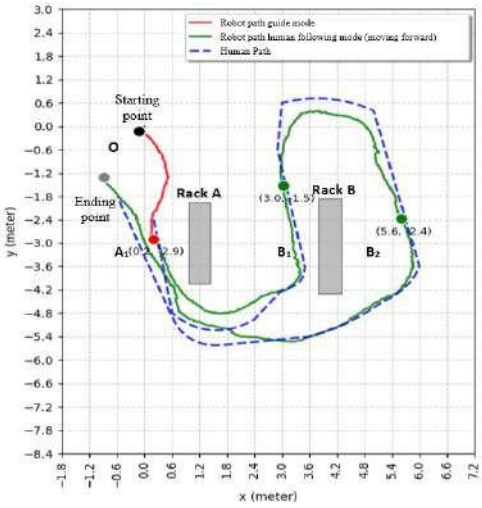


Fig. 13. Path comparison between human movement and robot movement on the A1A2B2O route

The system has a high level of functional integration, shown by the seamless switch between waypoint-based navigation, human-following (forward and backward), and stopping via real-time hand signal commands. This overcomes the challenge of control fragmentation common in-service robot architectures, where transitions between modes often require GUI intervention or operational pauses, as also noted in The Human-Following Strategy for Mobile Robots in Mixed Environments, as reported by Van Toan *et al.,* [36]. In the proposed architecture, the cue-based control is directly integrated into the Unified Perception Engine, making the workflow more unified and dynamic. In a

retail context, the human-robot interaction is more responsive as the system forms an efficient perception-to-control loop.

3.7 System Limitations

The tests demonstrate reliable performance, however, there are still limitations. First, the system is sensitive to lighting variations as it relies on a monocular RGB camera. YOLOv11-Pose model's detection confidence may be degraded due to fluctuation or poor lighting in supermarket aisles. Second, complete target loss may happen as there may be prolonged full-body occlusion caused by other shoppers or physical store structures. Kalman Filter mitigates occlusions effectively, the momentary occlusions though. Third, at a slightly lower precision (0.87), the recorded paper gesture sometimes causes misclassification as open-hand silhouettes can visually blend with complex background objects. Finally, the computational constraints of the NVIDIA Jetson Orin Nano limits the ability of the system to run more complex, multi-scale models concurrently alongside dense navigation algorithms without experiencing increased inference latency; although it provides adequate edge AI capabilities for the current tasks.

4. Conclusion

Successfully, this research develops a functional supermarket robot prototype. A design that offers an intuitive and reliable hand gesture-based control system. The success development of the model is a result of the systemic integration of several key components: an efficient YOLOv11-Pose model-based Unified Perception Engine, a Hybrid Tracking Protocol combining AprilTag for target acquisition and Kalman Filter for robust tracking, and an accurate monocular distance estimation method. Quantitatively, the system is valid, with a model F1Score above 0.92, a real-time inference speed of 14 FPS on edge devices, a distance estimation MAPE below 5.3%, and an average cue detection confidence value higher than 0.83. Robot can consistently identify the user. Moreover, it can validate hand gesture commands and respond to movement stably on a simulated supermarket trajectory. This includes forward and backward maneuvers as well as flexible mode transitions. Therefore, this research proposes an integrated and effective architecture for realizing an efficient, intuitive, human-following robot for application in dynamic retail environments.

In a simulated setting, the prototype demonstrates reliable performance. However, in actual supermarkets, scalability and robustness challenges remain. Different from the simulated setting, real-world retail spaces feature dense crowds, highly variable lighting, and narrow, and dynamic aisles. These real features test the limits of purely monocular vision-based perception. Hence, it is expected that future research considers system robustness. This can be achieved by combining the RGB camera data with low-cost complementary sensors, such as ultrasonic sensors or 2D LiDAR. The combination offers active obstacle avoidance without compromising the economic viability of the architecture. In addition, the scalability challenges can be addressed by exploring advanced mapping algorithms and multi-robot coordination systems. These improvements will ensure that a fleet of autonomous trolleys can seamlessly and safely operate alongside hundreds of shoppers in large-scale commercial deployments.

References

[1] Thompson, C*., et al.* "Changes to Household Food Shopping Practices during the COVID-19 Restrictions: Evidence from the East of England." *Health & Place 78* (2022): 102906. https://doi.org/10.1016/j.healthplace.2022.102906

[2] Wang, S., Y. Ye, B. Ning, J. H. Cheah, and X. J. Lim. "Why Do Some Consumers Still Prefer In-Store Shopping? An Exploration of Online Shopping Cart Abandonment Behavior." *Frontiers in Psychology* 12 (2022): 1–14. https://doi.org/10.3389/fpsyg.2021.829696

[3] Purwantono, H. Y., A. A. S. Gunawan, H. Tolle, M. Attamimi, and W. Budiharto. "A Literature Review: Feasibility Study of Technology to Improve Shopping Experience." *Procedia Computer Science* 179 (2021): 468–479. https://doi.org/10.1016/j.procs.2021.01.030

[4] Vikash Ranjan, D. L. M., and Shafaq Akhtar. "Study on Consumer Satisfaction and Loyalty Towards." International Journal of Advanced Multidisciplinary Research 10, no. 1 (2023): 89–99. https://doi.org/10.22192/ijamr.2023.10.01.009

[5] Ramzan, A., R. Mustafa, Z. Rehman, S. Suleman, M. Noor, and R. Bibi. "Robotrolley: Customer Following Trolley (CFT)." *In Proceedings of the 17th International Conference on Open Source Systems and Technologies (ICOSST 2023),* 1–6. 2023. https://doi.org/10.1109/ICOSST60641.2023.10414202

[6] Tsai, T. H., and C. H. Yao. "A Robust Tracking Algorithm for a Human-Following Mobile Robot*." IET Image Processing* 15, no. 3 (2021): 786–796. https://doi.org/10.1049/ipr2.12062

[7] Pătru, G. C., A. I. Pîrvan, D. Rosner, and R. V. Rughiniș. "Fiducial Marker Systems Overview and Empirical Analysis of Aruco, Apriltag and Cctag." UPB Scientific Bulletin, Series C: *Electrical Engineering and Computer Science* 85, no. 2 (2023): 49–62.

[8] Peñaloza-Aponte, D*., et al.* "Automated Entrance Monitoring to Investigate Honey Bee Foraging Trips Using Open-Source Wireless Platform and Fiducial Tags." *HardwareX* 20 (2024): e00609. https://doi.org/10.1016/j.ohx.2024.e00609

[9] Zachariae, A., F. Plahl, Y. Tang, I. Mamaev, B. Hein, and C. Wurll. "Human-Robot Interactions in Autonomous Hospital Transports." *Robotics and Autonomous Systems* 179 (2024): 104755. https://doi.org/10.1016/j.robot.2024.104755

[10] Reddy Kavya Sree, N., B. Anbarasu, and M. Luhitha. "Integration of LIDAR and Ultrasonic Sensor for MAV Collision Avoidance." In *Proceedings of the 5th International Conference on Emerging Systems and Intelligent Computing (ESIC 2025),* 832–836. 2025. https://doi.org/10.1109/ESIC64052.2025.10962656

[11] Pinnamaraju, H. V., P. R. Kapu, A. N. Juturu, and B. Anbarasu. "Distance Estimation for Collision Avoidance of Micro Aerial Vehicles Using LiDAR Sensor." *In Proceedings of the International Conference on Automation, Computing, and Renewable Systems (ICACRS 2022),* 157–161. 2022. https://doi.org/10.1109/ICACRS55517.2022.10029233

[12] Suwandi, D. P., I. K. Wibowo, and M. M. Bachtiar. "Estimation of Ball Position Using Depth Camera for Middle Size Goalkeeper Robot." *In Proceedings of the International Electronics Symposium (IES 2022),* 374–379. 2022. https://doi.org/10.1109/IES55876.2022.9888575

[13] Mingozzi, A., A. Conti, F. Aleotti, M. Poggi, and S. Mattoccia. "Monitoring Social Distancing With Single Image Depth Estimation." *IEEE Transactions on Emerging Topics in Computational Intelligence 6*, no. 6 (2022): 1290–1301. https://doi.org/10.1109/TETCI.2022.3171769

[14] Herdianto, H., P. Sihombing, F. Fahmi, and T. Tulus. "Improving Object Distance Measurement Based on Mono Camera Using Magnification and YOLO Methods." In *Proceedings of the International Conference on Control, Automation, Electronics, Robotics, and Artificial Intelligence (CERIA 2024),* 1–6. 2024. https://doi.org/10.1109/CERIA64726.2024.10914711

[15] Wicaksana, Y. D. T. T., A. Hendriawan, and N. Tamami. "Human Target Distance Estimation System Using Mono-camera On Human-Following Mobile Robot." *In Proceedings of the International Electronics Symposium (IES 2022), 349*–354. 2022. https://doi.org/10.1109/IES55876.2022.9888331

[16] Alhmiedat, T*., et al.* "A SLAM-Based Localization and Navigation System for Social Robots: The Pepper Robot Case." *Machines* 11, no. 2 (2023): 1–17. https://doi.org/10.3390/machines11020158

[17] Zhang, G., J. Yin, P. Deng, Y. Sun, L. Zhou, and K. Zhang. "Achieving Adaptive Visual Multi-Object Tracking with Unscented Kalman Filter." *Sensors* 22, no. 23 (2022): 1–18. https://doi.org/10.3390/s22239106

[18] Rasheed, A. F., and M. Zarkoosh. "YOLOv11 Optimization for Efficient Resource Utilization." *Journal of Supercomputing* 81, no. 9 (2025). https://doi.org/10.1007/s11227-025-07520-3

[19] Liu, L., A. Kurban, and Y. Liu. "Improved YOLOv11pose for Posture Estimation of Xinjiang Bactrian Camels." *International Journal of Advanced Computer Science and Applications* 15, no. 12 (2024): 364–371. https://doi.org/10.14569/IJACSA.2024.0151239

[20] Petković, M., and I. Vujović. "Distance Estimation Approach for Maritime Traffic Surveillance Using Instance Segmentation." *Journal of Marine Science and Engineering* 12, no. 1 (2024). https://doi.org/10.3390/jmse12010078

[21] Herbaz, N., H. El Idrissi, and A. Badri. "Deep Learning Empowered Hand Gesture Recognition: Using YOLO Techniques." In *Proceedings of the 14th International Conference on Intelligent Systems and Theoretical Applications (SITA 2023),* 1–7. 2023. https://doi.org/10.1109/SITA60746.2023.10373734

[22] Zhou, L., C. Du, Z. Sun, T. L. Lam, and Y. Xu. "Long-Range Hand Gesture Recognition via Attention-based SSD Network." *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2021),* 1832–1838. 2021. https://doi.org/10.1109/ICRA48506.2021.9561189

[23] Liu, L., D. Du, Y. Sun, and Y. Li. "SFMW-YOLO: A Lightweight Metal Casting Surface Defect Detection Method Based on Modified YOLOv8s." *Expert Systems with Applications 287* (2025): 128170. https://doi.org/10.1016/j.eswa.2025.128170

[24] Islam, M. D*., et al.* "Towards Real-Time Weed Detection and Segmentation with Lightweight CNN Models on Edge Devices." Computers and Electronics in Agriculture 237 (2025): 110600. https://doi.org/10.1016/j.compag.2025.110600

[25] Salimi, S., J. P. Queralta, and T. Westerlund. "Hyperledger Fabric Blockchain and ROS 2 Integration for Autonomous Mobile Robots." In *Proceedings of the IEEE/SICE International Symposium on System Integration (SII 2023).* 2023. https://doi.org/10.1109/SII55687.2023.10039326

[26] Gutierrez-Flores, P. A., and R. Bachmayer. "Concept Development of a Modular System for Marine Applications Using ROS2 and Micro-ROS." In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Symposium (AUV 2022).* 2022. https://doi.org/10.1109/AUV53081.2022.9965867

[27] Li, H., et al. "Optimizing Edge-Enabled System for Detecting Green Passion Fruits in Complex Natural Orchards Using Lightweight Deep Learning Model." *Computers and Electronics in Agriculture* 234 (2025): 110269. https://doi.org/10.1016/j.compag.2025.110269

[28] Pan, S., *et al*. "A Lightweight Robust RGB-T Object Tracker Based on Jitter Factor and Associated Kalman Filter." *Information Fusion* 117 (2025): 102842. https://doi.org/10.1016/j.inffus.2024.102842

[29] Kumar, A., R. Vohra, R. Jain, M. Li, C. Gan, and D. K. Jain. "Correlation Filter Based Single Object Tracking: A Review." *Information Fusion* 112 (2024): 102562. https://doi.org/10.1016/j.inffus.2024.102562

[30] Schramm, S., J. Rangel, D. A. Salazar, R. Schmoll, and A. Kroll. "Target Analysis for the Multispectral Geometric Calibration of Cameras in Visual and Infrared Spectral Range*." IEEE Sensors Journal* 21, no. 2 (2021): 2159–2168. https://doi.org/10.1109/JSEN.2020.3019959

[31] Shih, C. L., W. C. Huang, I. T. Anggraini, Y. Xiao, N. Funabiki, and C. P. Fan. "Performance Comparison between OpenPose and TRT-Pose for Self-Practice Yoga on Embedded GPU Platform." *In Proceedings of the 2023 IEEE International Conference on Consumer Electronics (ICCE-Asia 2023*), 1–4. 2023. https://doi.org/10.1109/ICCE-Asia59966.2023.10326363

[32] Chen, J., S. Bai, G. Wan, Y. Li, J. Li, and Z. Cheng. "Research on Defect Detection Method of Automotive Running Lights Based on Keypoint Identification." In *Proceedings of the 36th Chinese Control and Decision Conference (CCDC 2024*), 5220–5225. 2024. https://doi.org/10.1109/CCDC62350.2024.10588328

[33] Duarte, R. P., C. A. Cunha, and J. C. Cardoso. "Automatic Camera Calibration Using a Single Image to Extract Intrinsic and Extrinsic Parameters." *International Journal of Intelligent Systems and Applications in Engineering (IJISAE*) 2024, no. 3 (2024): 1766–1778. https://www.ijisae.org/index.php/IJISAE/article/view/5586

[34] Tsai, T. H., and C. L. Lee. "Equipped with Monocular Depth Estimation and Intelligent Wake-Up Vision Based Tracking System for a Human-Following Mobile Robot*." In Proceedings of the 2024 IEEE International Conference on Multimedia Expo Workshop (ICMEW 2024),* 1–2. 2024. https://doi.org/10.1109/ICMEW63481.2024.10645403

[35] Yong, C. L., B. Hoe Kwan, D. W. K. Ng, and H. Seng Sim. "Human Tracking and Following Using Machine Vision on a Mobile Service Robot." In *Proceedings of the 2022 IEEE 10th Conference on Systems and Process Control (ICSPC 2022),* 274–279. 2022. https://doi.org/10.1109/ICSPC55597.2022.10001803

[36] Van Toan, N., M. Do Hoang, P. B. Khoi, and S. Y. Yi. "The Human-Following Strategy for Mobile Robots in Mixed Environments." *Robotics and Autonomous Systems* 160 (2023): 104317. https://doi.org/10.1016/j.robot.2022.104317