

AI-Based Autonomous Warehouse Robot Car: Design, Integration, and Experimental Evaluation of an Edge-AI Sorting Platform

Almutasim Billa Alanazi, Abdullah Bamufleh, Mohammed Arnaout

Department of Electrical Engineering, Faculty of Engineering, Islamic University of Madinah, Madinah, Saudi Arabia

Abstract- Warehouse automation is increasingly important for improving operational efficiency, reducing sorting errors, and supporting high-throughput logistics. This paper presents the design, implementation, and experimental evaluation of an AI-based autonomous warehouse robot car for real-time color-based sorting of uniformly sized boxes in an indoor warehouse environment. The robot integrates a dual-camera perception architecture, an ONNX-optimized YOLO object-detection model, ArUco marker-based localization, a Raspberry Pi 5 embedded processor, four Mecanum wheels for omnidirectional mobility, ultrasonic sensing for final approach control, a servo-driven gripper, and a Flask-based real-time monitoring interface. Autonomous task execution is organized using a finite-state machine that sequences Search, Align, Approach, Pick, Deliver, and Exit states, while a deterministic priority rule (Red > Blue > Green) defines the order of box handling. Experimental testing was performed in an approximately 3 m × 3 m simulated warehouse with multiple runs and varying numbers of red, blue, and green boxes. The reported results showed object-detection accuracy of 85%–95%, alignment accuracy of approximately 92%, pick-and-place success of 91%–95% with more than 95% success in most test conditions, and delivery accuracy of approximately 97%. The average detection-to-delivery time was approximately 20 s per box, system latency was below 100 ms, and the priority-sorting logic achieved 100% correctness with no cross-color delivery errors. The findings demonstrate the effectiveness of integrating edge AI, marker-based localization, omnidirectional locomotion, robotic manipulation, and real-time monitoring on a low-cost embedded platform.

Keywords— warehouse automation; autonomous mobile robot; YOLO; ArUco markers; Mecanum wheels; Raspberry Pi 5; edge AI; robotic sorting; finite-state machine.

I. INTRODUCTION

The rapid growth of global supply chains and e-commerce has increased the demand for warehouse systems that can operate with high efficiency, accuracy, and consistency. Conventional manual sorting remains labor-intensive, time-consuming, and vulnerable to human fatigue and error. These development considerations have accelerated the adoption of autonomous mobile robots (AMRs) as an enabling technology for Industry 4.0 logistics [1], [2]. Industrial warehouse automation can provide excellent throughput, but the associated infrastructure, centralized control, sensing hardware, and integration cost can make such solutions difficult to adopt in small and medium-sized facilities.

This work was motivated by the need for a practical and accessible platform that combines modern robotics and artificial intelligence without relying on expensive industrial controllers or cloud computation. The proposed robot performs end-to-end autonomous sorting of uniformly sized, color-coded boxes. It detects objects, selects the next target according to a predefined priority, aligns with the target, approaches and grips it, navigates to a designated destination, releases the object, and repeats the sequence until the required sorting mission has been completed.

The system combines several engineering domains within a single mobile platform. ArUco markers are used for accurate indoor localization [3]. A finite-state-machine architecture provides structured and predictable sequencing of robot behavior [4]. Four Mecanum wheels provide holonomic movement and

enable lateral, diagonal, and rotational motion in confined spaces [5]. A real-time monitoring interface provides visibility into the robot's operating state, mission progress, and system behavior, consistent with the role of connected interfaces in IoT-enabled robotic systems [6].

The objective of this paper is to convert the study into a research-oriented presentation focused on the technical architecture, design decisions, implementation methodology, test procedure, experimental results, system development considerations, safety considerations, and future research opportunities. The contribution is primarily systems-oriented: instead of investigating a single algorithm in isolation, the work demonstrates the integration of perception, localization, decision-making, locomotion, manipulation, and monitoring on a low-cost Raspberry Pi 5 platform.

II. BACKGROUND AND RELATED WORK

1. Warehouse automation and AMRs

Warehouse automation has evolved rapidly in response to rising logistics volumes and expectations for fast order fulfillment. AMRs and automated material-handling systems have become important tools for transporting and sorting items with reduced human intervention. Many industrial solutions rely on centralized control and infrastructure-supported navigation, including magnetic tracks, QR codes, and predefined routes. Although such systems can be highly reliable and provide strong throughput, they may require substantial installation cost and may be inflexible when warehouse layouts change [10], [11].

Alternative navigation strategies differ significantly in complexity and flexibility. Line-following robots are attractive because of their simplicity, low cost, and predictable motion, but their routes are physically constrained and difficult to modify. SLAM-based platforms provide substantially greater flexibility because they can localize and map unknown environments; however, the required sensors and computation can be expensive for low-cost embedded systems [12], [13]. The scope of the present prototype therefore excludes advanced

SLAM to maintain computational simplicity and focus on the integration of core warehouse-sorting functions [8], [9].

2. Comparison of candidate sorting approaches

The design study compared line-following, SLAM-based, industrial, and the proposed free-moving system in terms of complexity, cost, flexibility, performance, real-time capability, and educational suitability. The free-moving vision-based design was selected because it provided a practical compromise: greater flexibility than line following, lower cost and computational demand than SLAM or industrial systems, and sufficient real-time performance for the intended controlled environment.

Table 1. Comparison of warehouse sorting approaches used in the system-design evaluation.

Criterion	Line-following	SLAM - based	Industrial systems	Proposed system
Complexity	Low	High	Very high	Medium
Cost	Low	High	Very high	Low
Flexibility	Limited	High	Limited	High
Performance	Moderate	High	Very high	High
Real-time capability	Yes	Sometimes	Yes	Yes
Educational suitability	High	Low	Low	High

3. Computer vision and localization

Computer vision allows the robot to identify objects and interpret the operating environment. The YOLO family of single-stage detectors is widely used in real-time robotics because it provides a favorable compromise between inference speed and detection accuracy [14]. For embedded deployment, model conversion and optimization can reduce latency and resource demand. The proposed system therefore uses a trained YOLO model exported to ONNX for execution on the Raspberry Pi 5 [15].

ArUco fiducial markers provide a comparatively lightweight method for pose estimation and localization in structured indoor environments [16], [17]. The proposed system adopts a dual-camera architecture in which one camera is dedicated to object detection while another is used for marker

detection and localization. This separation is intended to reduce the compromises of a single-camera design and improve both object-recognition reliability and positioning precision [18].

4. Locomotion and embedded control

Differential-drive mobile robots are simple and widely used, but they have limited lateral maneuverability. Mecanum-wheel systems provide true omnidirectional motion, allowing the chassis to translate laterally and diagonally and to rotate in place without changing its general heading [19]. This capability is advantageous in warehouse environments where aisle space is limited and precise position correction is required.

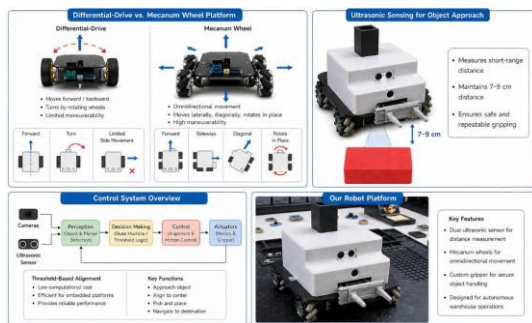


Figure 3. Comparison between differential-drive and Mecanum-wheel platforms, illustrating the increased maneuverability of the selected omnidirectional configuration.

The design process also considered multiple control strategies, including rule-based and threshold-based methods, PID control, and machine-learning-based approaches. For this prototype, threshold-based alignment was selected because it offers a practical balance between computational efficiency and acceptable positioning performance on an embedded platform [20]. Public datasets and robotic simulation environments such as Gazebo and Webots are commonly used for safe development and validation before physical deployment [21].

III. SYSTEM DESIGN AND METHODOLOGY

1. Overall architecture

The system is organized into four interacting layers: perception, processing, actuation, and monitoring.

The perception layer contains two cameras and an ultrasonic sensor. The processing layer is implemented on a Raspberry Pi 5 with 4 GB RAM and runs the YOLO-ONNX object detector, ArUco detection, distance processing, state-machine logic, logging, and robot-control scripts. The actuation layer contains the motor drivers, four DC gear motors, Mecanum wheels, and the MG996R servo gripper. The monitoring layer is a Flask-based web dashboard that displays the live camera feed, robot state, mission progress, and operating events.

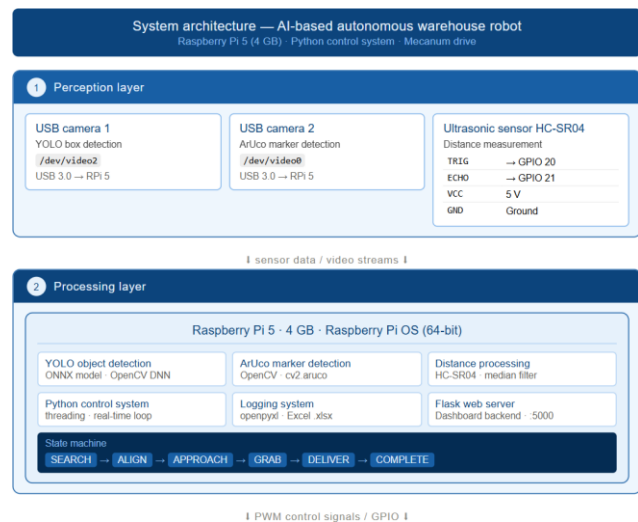


Figure 6. Overall system architecture showing the perception, processing, actuation, and monitoring layers and the principal data/control flows.

2. Autonomous task workflow

After the user enters the required number of boxes for each color class, the robot operates autonomously. The finite-state machine follows the sequence Search → Align → Approach → Pick → Deliver → Exit. During Search, the vision subsystem scans the environment for candidate boxes. When more than one box is available, the controller applies the priority order Red > Blue > Green. The Align state centers the selected target using image feedback. In Approach, the robot moves toward the target while the ultrasonic sensor monitors the remaining distance. At the desired grasping distance, the gripper is actuated. The robot then transitions to Deliver, identifies the appropriate ArUco marker associated with the carried box color, navigates to the corresponding zone, releases the box, and returns to the search sequence.

This structure was chosen to make robot behavior deterministic and easy to debug. Each subsystem performs a clearly defined function, while the finite-state controller coordinates their interaction. Fallback behavior is also possible: temporary loss of visual information or alignment can cause the controller to return to Search or Align rather than continuing with an unsafe approach.

3. Hardware platform

The hardware platform is a compact four-wheel mobile robot. Four Mecanum wheels are mounted at the chassis corners and driven by DC gear motors through motor-driver modules. The vision system includes a general-purpose webcam for box detection and a high-resolution Arducam module for ArUco marker localization. An HC-SR04 ultrasonic sensor is mounted at the front of the robot to measure short-range distance during the approach phase. The manipulation mechanism is a servo-actuated gripper designed for simple and repeatable pick-and-place handling of the uniformly sized boxes.

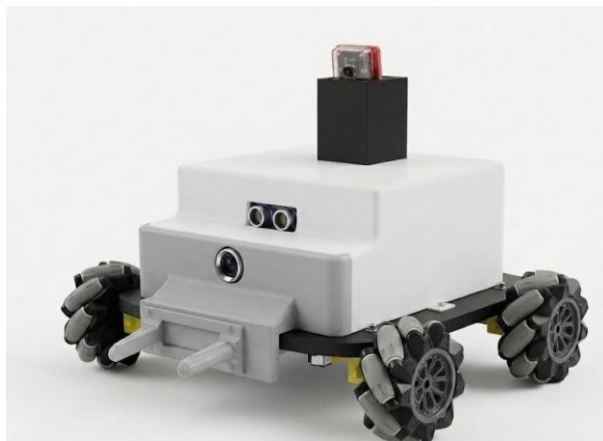


Figure 7. Robot kit and hardware elements used in the autonomous warehouse robot prototype.

4. Mechanical design and chassis construction

The chassis was modeled in Fusion 360 to establish component dimensions, structural relationships, and assembly locations before fabrication. The mechanical design sought a balance among rigidity, low mass, manufacturability, and accessibility for maintenance. Acrylic sheets were used as primary structural material, while 3D-printed elements were employed for customized sensor mounts, supports,

and other components. This combination provides structural strength and design flexibility at relatively low weight [22].

The four Mecanum wheels are arranged in the standard X-configuration. This geometry enables forward and reverse translation, lateral movement, diagonal motion, and rotation in place [19], [23]. The placement of the Raspberry Pi, battery pack, motor drivers, cameras, and gripper was also considered to reduce component interference and simplify wiring. Heavier components were positioned low in the chassis to reduce the center of gravity and improve stability during motion.

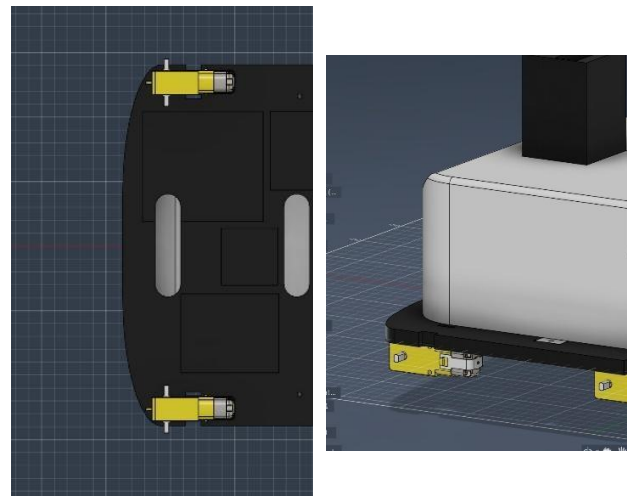


Figure 8. CAD renderings of the robot chassis, component arrangement, and Mecanum-wheel mounting configuration.

5. Electrical system and motor control

The electrical architecture was designed to provide regulated power, motor control, sensor interfacing, and reliable operation of the processing hardware. The electrical design uses a rechargeable lithium-ion battery supply rated at 11.1 V and 5 A. A voltage-regulation stage supplies the Raspberry Pi and low-power electronics at approximately 5.2 V [24]. The four DC motors are driven through motor-driver modules that receive digital direction commands and pulse-width-modulation signals from the Raspberry Pi.

The Raspberry Pi serves as the central controller and interfaces with the motion system, cameras,

ultrasonic sensor, and servo gripper through GPIO. The ultrasonic sensor is triggered through GPIO and returns distance measurements for the final approach. The servo is controlled by PWM to coordinate opening and closing of the gripper. The electrical design emphasizes secure wiring, common grounding, and appropriate separation of high-current motor loads from sensitive processing and sensing components.



Figure 9. Kinematic representation of the four-Mecanum-wheel robot, including wheel arrangement, roller orientation, and achievable motion directions.

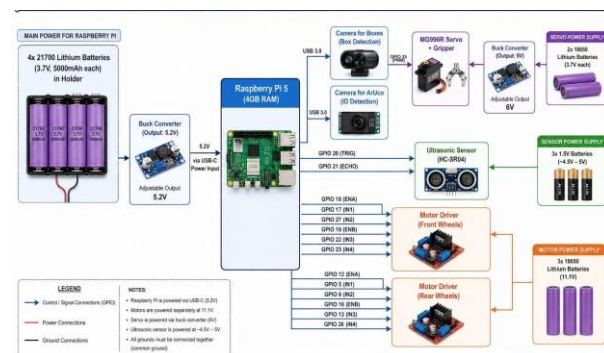


Figure 10. Electrical-system diagram showing power distribution, motor-driver connections, Raspberry Pi GPIO/PWM interfaces, cameras, ultrasonic sensing, and gripper control.

IV. EXPERIMENTAL SETUP AND EVALUATION PROCEDURE

System testing was performed in an indoor area of approximately 3 m × 3 m configured to represent a small warehouse. Three ArUco markers, identified as

IDs 1, 2, and 3, were placed on the walls at predetermined drop-off zones. Uniformly sized red, blue, and green boxes were positioned at varying locations in the search area. The robot was tested over multiple runs with varying numbers of boxes in each color class rather than a single fixed workload. All reported experiments were performed under normal indoor lighting.

The evaluation considered the complete perception-to-delivery cycle. Vision-system performance was assessed through the ability of the YOLO model to recognize and classify boxes and the ability of the ArUco camera to identify the correct destination marker. Navigation and manipulation were assessed through target alignment, approach-distance control, box pickup, transport, and delivery. Timing evaluation measured the overall detection-to-delivery duration for a single box and observed the effect of multi-box workloads. The experiments also assessed the correctness of the priority rule and the stability of state-machine transitions.

The evaluation environment was structured to provide repeatable and consistent testing of the integrated robotic system. Controlled lighting, uniformly sized objects, and predefined drop-off zones enabled focused assessment of the complete perception-to-delivery sequence. This experimental design supported systematic evaluation of object detection, alignment, navigation, manipulation, priority-based decision-making, and real-time monitoring across multiple autonomous runs.

V. EXPERIMENTAL RESULTS

1. Vision-system Performance

The custom-trained YOLO model, converted to ONNX format for execution on the Raspberry Pi 5, demonstrated reliable real-time detection and color classification. The experimental results show an object-detection accuracy range of 85%–95%. Detection performance varied with the distance between the camera and the box, the viewing angle, and lighting conditions. The dual-camera architecture was effective in separating box detection from destination localization, and ArUco

marker detection remained particularly stable during the reported experiments.

2. Alignment and pick-and-place performance

Target-alignment accuracy was approximately 92%. During the final approach, the ultrasonic sensor maintained the desired gripping distance of approximately 7–9 cm. The experimental evaluation demonstrates a pick-and-place success rate of 91%–95%, while also noting that pickup success exceeded 95% in most tested conditions. The combination of vision-based alignment, short-range distance feedback, and servo actuation therefore provided repeatable object handling in the controlled environment.

3. Navigation and delivery

ArUco marker-based localization provided high delivery precision. The robot correctly associated each box color with its corresponding destination marker and achieved approximately 97% delivery accuracy. This result indicates that the marker-based navigation strategy was suitable for the controlled prototype environment and that the dual-camera system provided sufficiently stable localization for repeated pick-and-deliver cycles.

4. Timing, priority logic, and stability

In single-box experiments, the average time from initial detection to successful delivery was approximately 20 s. This interval included detection, alignment, approach, picking, navigation, and release. Reported system latency remained below 100 ms. When multiple boxes were present, total mission time increased approximately in proportion to the number of boxes because the tasks were executed sequentially.

The priority-based sorting policy Red > Blue > Green achieved 100% correctness in the reported test runs. No cross-color delivery errors were observed. State-machine transitions were smooth and no critical system failures were reported across the repeated experiments. Minor degradation was observed when battery voltage decreased, resulting in lower motor speed and increased alignment time.

Table 2. Summary of the performance metrics obtained for the autonomous warehouse robot.

Performance metric	Reported value	Interpretation within test scope
Object-detection accuracy	85%–95%	High real-time detection performance across tested conditions
Alignment accuracy	≈92%	Adequate for controlled pick approach
Pick-and-place success	91%–95%; >95% in most conditions	Reliable repeatable manipulation performance
Delivery accuracy	≈97%	Stable ArUco-based destination localization
Average time per box	≈20 s	Complete detection-to-delivery sequence
System latency	<100 ms	Supports real-time embedded control
Priority sorting correctness	100%	No cross-color delivery errors

VI. RISK ANALYSIS AND SAFETY CONSIDERATIONS

Risk analysis was included as an explicit part of the engineering design because the system combines physical motion, electrical power, robotic manipulation, and AI-based perception. The safety analysis identifies mechanical, electrical, collision/navigation, vision, operational, and human–robot interaction hazards. The overall risk-management approach is aligned with ISO 12100 risk-assessment principles and ISO 10218 robotic-system safety guidance [26], [27].

Mechanical hazards arise from moving wheels, motor shafts, and the gripper. Electrical hazards include overheating, short circuits, voltage instability, over-discharge, and other risks associated with lithium-ion batteries, converters, and high-current motor drivers. Collision risk can result from software errors, sensor errors, or the lack of advanced obstacle avoidance. Vision failures include YOLO misclassification or temporary loss of ArUco tracking, while manipulation errors can lead to dropped objects.

The safety analysis uses a qualitative risk matrix based on likelihood categories from Rare to Almost Certain and severity categories from Negligible to Catastrophic. Risk levels are then classified as Low, Medium, High, or Extreme. This structured approach supports the prioritization of mitigation measures and is consistent with the probability-and-severity framework described in ISO 12100.

Implemented safety measures include a software watchdog that stops the motors if the main control process freezes, a hardware emergency-stop function, ultrasonic-based proximity protection, separation of high-current and sensitive-electronics power domains, LED status indicators, and intentional speed development consideration during search and alignment. The state machine also provides fallback states so the robot can recover from temporary vision loss or misalignment rather than continuing an unsafe maneuver.

VII. DISCUSSION AND ENGINEERING SIGNIFICANCE

1. System strengths

The experimental results demonstrate that the proposed architecture can coordinate perception, localization, manipulation, and mobility on a low-cost embedded computer. The dual-camera design is a major system-level strength because it separates object recognition and localization tasks, reducing sensing conflicts and allowing each camera to be used for a task-specific purpose. The ONNX-optimized YOLO model further demonstrates that real-time AI inference can be implemented on the Raspberry Pi 5 without requiring cloud computation. The Mecanum-wheel configuration also contributes to the platform's usefulness. Omnidirectional motion permits lateral correction and rotation in place, reducing the space required for repositioning. This is particularly relevant to warehouse-like environments where the robot must align with boxes and destination zones in confined areas. The deterministic finite-state-machine implementation provides predictable behavior and simplifies debugging, while the Flask dashboard increases observability during testing and demonstration.

2. Performance interpretation and engineering insights

The experimental program was designed around a structured indoor warehouse scenario using uniformly sized, color-coded objects and predefined destination markers. This configuration enabled repeatable evaluation of the complete autonomous sorting cycle and allowed the performance of the perception, localization, locomotion, manipulation, and decision-making modules to be assessed consistently. The marker-based localization strategy provided a practical and computationally efficient foundation for accurate delivery within the experimental environment.

Across the experimental runs, the vision subsystem achieved detection accuracy in the reported range of 85%–95%, while the alignment subsystem achieved approximately 92% accuracy. The results demonstrate effective real-time perception across the tested object positions and viewing conditions. The integrated power and motion architecture also supported repeated autonomous missions, highlighting the importance of coordinated sensing, processing, actuation, and energy management in mobile robotic platforms.

An important engineering outcome of the project is the successful integration of perception, state logic, motion, gripper actuation, and monitoring into a unified autonomous platform. Iterative testing supported coordinated operation across these subsystems. The dual-camera arrangement demonstrates the value of task separation in embedded perception systems, while the integrated power architecture illustrates the role of systematic energy management in sustaining reliable mobile-robot operation.

VIII. OPPORTUNITIES FOR FURTHER DEVELOPMENT

1. Multi-robot coordination

The demonstrated single-robot architecture provides a strong foundation for extending the system toward collaborative multi-robot warehouse operation. Further development can investigate

centralized or decentralized fleet management, task-allocation algorithms based on robot availability, battery level, and proximity to targets, real-time communication through frameworks such as MQTT or ROS 2, and coordinated inter-robot navigation. Multi-robot coordination is an important research direction in modern Industry 4.0 automation [29].

2. Advanced sensing and navigation

The established 2D vision, marker-based localization, and threshold-based alignment architecture can serve as a platform for investigating additional sensing and navigation capabilities. Further development may integrate 3D depth cameras, stereo vision, or LiDAR to enrich environmental perception and object localization. SLAM can extend autonomous mapping and localization capabilities [8], [28], while LiDAR-based perception offers an additional sensing modality for warehouse navigation [9]. Closed-loop alignment strategies, including PID or model-predictive control, also provide promising directions for increasingly precise motion control.

3. Improved AI and warehouse integration

The perception system could be retrained on a larger and more diverse dataset containing illumination changes, partial occlusion, and more varied object appearances [14], [30]. Further development includes possible use of model pruning or edge-acceleration tools to further reduce inference time, along with investigation of lightweight transformer or hybrid vision architectures. Additional warehouse functions could include barcode or QR-code identification, automatic box-size detection, adaptive gripping-force control, real-time inventory tracking, integration with a Warehouse Management System, and dynamic task reprioritization.

Power autonomy is another important extension. A battery-management system could estimate state of charge, predict remaining mission time, and trigger automatic docking for recharging. Cloud-edge architectures may also be useful for remote monitoring and software updates while retaining local execution of safety-critical perception and control.

IX. CONCLUSION

This paper presented a detailed journal-style account of the AI-Based Autonomous Warehouse Robot Car developed as a fully integrated sorting prototype. The robot combines a Raspberry Pi 5, a custom YOLO detector optimized in ONNX format, a dual-camera vision system, ArUco marker localization, Mecanum-wheel omnidirectional mobility, ultrasonic-assisted approach control, a servo-based gripper, finite-state-machine sequencing, priority-based sorting, and a Flask monitoring dashboard.

Testing in an approximately 3 m × 3 m simulated warehouse demonstrated object-detection accuracy of 85%–95%, approximately 92% alignment accuracy, 91%–95% pick-and-place success with more than 95% success in most tested conditions, approximately 97% delivery accuracy, average processing of approximately 20 s per box, system latency below 100 ms, and 100% correctness of the Red > Blue > Green priority policy. The results support the feasibility of low-cost edge-AI warehouse robotics for structured environments.

The completed prototype demonstrates a coherent and effective integration of edge-AI perception, marker-based localization, omnidirectional mobility, robotic manipulation, deterministic control, and real-time monitoring. The architecture provides a strong technical foundation for continued development in multi-robot coordination, SLAM, 3D sensing, enhanced AI models, closed-loop control, inventory-system integration, and autonomous energy management.

REFERENCES

1. A. K. Grover and M. H. Ashraf, "Leveraging Autonomous Mobile Robots for Industry 4.0 Warehouses: A Multiple Case Study Analysis," *The International Journal of Logistics Management*, vol. 35, no. 4, pp. 1168–1199, 2024. <https://doi.org/10.1108/IJLM-09-2022-0362>
2. M. Soori, B. Arezoo, and R. Dastres, "Autonomous Mobile Robots in Logistics and

- Supply Chain Management of Industry 4.0, a Review," *Journal of Mechanical and Industrial Engineering*, 2024. <https://www.researchgate.net/publication/399713700>
3. A. Sampathkrishna, "ArUco Marker Based Localization and Node Graph Approach to Mapping," *arXiv preprint*, Aug. 2022. <https://arxiv.org/abs/2208.09355>
4. M. Ben-Ari and F. Mondada, "Finite State Machines," in *Elements of Robotics*, Springer, 2018, pp. 55–61. https://link.springer.com/chapter/10.1007/978-3-319-62533-1_4
5. O. Bozovic et al., "Kinematic Analysis of Omnidirectional Mecanum Wheeled Robot," *Int. Journal of Engineering and Applied Physics*, vol. 3, no. 1, pp. 634–644, Jan. 2023. <https://ijeap.org/ijeap/article/download/122/109/1025>
6. A. Lojka et al., "Internet of Robotic Things: Current Technologies, Challenges, Applications, and Future Research Topics," *PMC – Sensors*, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11820596/>
7. M. A. Khalloof et al., "Development of a Low-Cost Ground Robot for Real-Time ArUco Marker Localization Using Lightweight Deep Learning Models on an Edge Device," *Smart Agricultural Technology*, 2026. <https://www.sciencedirect.com/science/article/pii/S277237552600122X>
8. A. B. Abaspur Kazerouni et al., "A Survey of State-of-the-Art on Visual SLAM," *Expert Systems with Applications*, vol. 205, p. 117734, 2022. <https://doi.org/10.1016/j.eswa.2022.117734>
9. R. Keith and H. M. La, "Review of Autonomous Mobile Robots for the Warehouse Environment," *arXiv preprint*, Jun. 2024. <https://arxiv.org/html/2406.08333v1>
10. K. Fragapane et al., "Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021. <https://doi.org/10.1016/j.ejor.2021.01.019>
11. A. M. Nasser et al., "AGV and Industry 4.0 in Warehouses: A Comprehensive Analysis of Existing Literature and an Innovative Framework for Flexible Automation," *The International Journal of Advanced Manufacturing Technology*, 2024. <https://link.springer.com/article/10.1007/s00170-024-14127-0>
12. D. Niermann, C. Petzoldt, and M. Freitag, "Intuitive and Flexible Process Control for Autonomous Mobile Robots: A Case Study in a Large Logistics Enterprise," *Procedia CIRP*, vol. 120, pp. 153–158, 2023. <https://doi.org/10.1016/j.procir.2023.08.028>
13. B. Bernardo, J. M. Sousa, and P. J. Gonçalves, "Survey on Robotic Systems for Internal Logistics," *Journal of Manufacturing Systems*, vol. 65, pp. 339–350, 2022. <https://doi.org/10.1016/j.jmsy.2022.09.014>
14. T. Gupta et al., "YOLO-ReT: Towards High Accuracy Real-Time Object Detection on Edge GPUs," *arXiv preprint*, 2021. <https://arxiv.org/pdf/2110.13713>
15. S. Mandal et al., "Efficient Edge Deployment of Quantized YOLOv4-Tiny for Aerial Emergency Object Detection on Raspberry Pi 5," *arXiv preprint*, 2025. <https://arxiv.org/html/2506.09300v1>
16. M. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Fiducial Markers for Pose Estimation: Overview and Performance Comparison," *Journal of Intelligent & Robotic Systems*, 2021. <https://doi.org/10.1007/s10846-020-01307-9>
17. E. Meng et al., "Enhancing Pose Estimation Using Multiple Graphical Markers with Spatial and Temporal Outlier Detection," *Applied Sciences*, vol. 14, no. 22, 2024. <https://www.mdpi.com/2076-3417/14/22/10225>
18. F. Mutsch et al., "Investigation of ArUco Marker Placement for Planar Indoor Localization," *arXiv preprint*, Sep. 2025. <https://arxiv.org/html/2509.17345v1>
19. L. Tagliavini et al., "Wheeled Mobile Robots: State of the Art Overview and Kinematic Comparison Among Three Omnidirectional Locomotion Strategies," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 57, 2022. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9589656/>

20. I. Bozovic et al., "Real-Time Navigation of Mecanum Wheel-Based Mobile Robot in a Dynamic Environment," *Heliyon*, vol. 10, 2024. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10982899/>
21. M. Rossander and H. Lideskog, "Design and Implementation of a Control System for an Autonomous Reforestation Machine Using Finite State Machines," *Forests*, vol. 14, no. 7, 2023. <https://www.mdpi.com/1999-4907/14/7/1340>
22. F. Y. Liang et al., "Design, Modeling, and Control of a New Multi-Motion Mobile Robot Based on Spoked Mecanum Wheels," *Micromachines* (MDPI), vol. 14, no. 6, 2023. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10204521/>
23. M. De Luca et al., "Hardware in the Loop Topology for an Omnidirectional Logistic Robot," *Symmetry* (MDPI), vol. 13, no. 6, p. 969, 2021. <https://pdfs.semanticscholar.org/1af7/83edb85fa9cce0f86cf9ceb6092ede48df3d.pdf>
24. D. McNulty et al., "A Review of Li-Ion Batteries for Autonomous Mobile Robots: Perspectives and Outlook for the Future," *Journal of Power Sources*, vol. 545, 2022. <https://doi.org/10.1016/j.jpowsour.2022.231943>
25. N. Promkaew et al., "Development of Autonomous Mobile Robot for Warehouse Applications," *International Journal of Mechanical Engineering and Robotics Research*, vol. 14, no. 3, 2025. <https://www.ijmerr.com/2025/IJMERR-V14N3-360.pdf>
26. ISO 12100:2010 – Safety of Machinery: General Principles for Design – Risk Assessment
27. ISO 10218-1:2011 – Robots and Robotic Devices: Safety Requirements for Industrial Robots, International Organization for Standardization, 2011. <https://www.iso.org/standard/51330.html>
28. S. Macenski et al., "The Marathon 2: A Navigation System," in *Proc. IEEE/RSJ IROS*, 2020. <https://arxiv.org/abs/2003.00368>
29. M. Gia Pham, "Applications of Industry 4.0 Technologies in Warehouse Management: A Systematic Literature Review," *Logistics* (MDPI), vol. 7, no. 2, 2023. <https://www.mdpi.com/2305-6290/7/2/24>
30. A. Comprehensive Evaluation of YOLO-based Detection on Edge Devices, "A Comprehensive Evaluation of YOLO-based Deer Detection Performance on Edge Devices," *arXiv preprint*, 2025. <https://arxiv.org/html/2509.20318v2>