Automated guided vehicle systems using AprilTag and ant colony optimization ESP32-based

Bryan Aurelius Rodney[1], *M. Ibrahim* Nur Hidayatullah[1], *Alberta* Bihaqil Hakiki[1], *Melani* Silvia Prameswari[1], *Nur* Vidia Laksmi B.[1*]

[1]Departement of Electrical Engineering, Faculty of Vocational Studies, Universitas Negeri Surabaya, Surabaya, Indonesia

Abstract. This paper presents the development of a prototype automated guided vehicle (AGV) using the ESP32 microcontroller, integrated with AprilTag fiducial markers for visual positioning and the Ant Colony Optimization (ACO). The aim is to design a low-cost AGV system capable of autonomous navigation in structured environments, such as container terminals. The AGV prototype employs an IP camera for real-time AprilTag detection, providing accurate localization data to the ESP32, which processes movement commands. The ACO algorithm is implemented to calculate the optimal route from the AGV's starting position to the target destination while avoiding obstacles. Experimental results demonstrate reliable AprilTag detection with a success rate exceeding 95% under proper lighting conditions. The use of ACO reduced overall travel distance by approximately 15% compared to manual routing methods. The combination of AprilTag and ACO provides an effective, low-cost solution for enhancing AGV navigation performance in industrial settings.

1 Introduction

The growing need for autonomous and efficient transportation systems is especially evident in logistics and manufacturing sectors, notably in container terminal operations where cargo handling demands high levels of accuracy and dependability. Automated Guided Vehicles (AGVs) are increasingly utilized to streamline internal logistics, reducing manual labour and enhancing overall operational efficiency [1-3]. Despite this, challenges such as precise localization and intelligent path planning continue to hinder the broader deployment of AGVs, particularly in cost-sensitive educational and experimental settings [4-6]. In parallel with these trends, this study is positioned within advances in artificial intelligence for robotics [7]. We incorporate a lightweight deep-learning model for image optimization encompassing learned deblurring, denoising, and super-resolution to fortify camera-based perception under variable illumination, motion blur, and occlusion. This AI-enhanced preprocessing improves the reliability of visual feature extraction while maintaining low computational overhead suitable for embedded platforms [8, 9].

One promising solution for localization is the use of AprilTag visual markers, which have demonstrated reliable performance even under diverse lighting conditions [10]. Simultaneously, nature-inspired algorithms like Ant Colony Optimization (ACO) have shown great potential in robotic navigation tasks, offering near-optimal solutions in complex. While prior research supports the effectiveness of AprilTag-based localization and ACO in separate contexts, environments by emulating the collective behavior of ant colonies. Their combined application on a low-cost AGV platform, particularly one based on the ESP32 microcontroller for use in container terminals, is still relatively unexplored [11].

Beyond ACO, the proposed framework is situated within a broader ecosystem of artificial-intelligence optimization methods that can complement or extend AGV performance. Population based metaheuristics such as Particle Swarm Optimization (PSO) etc. The applications include in many various field, machine learning, manufacturing, processing etc. [12,13]. This study seeks to design and implement a prototype AGV that integrates AprilTag-based positioning with ACO-driven path planning, offering a feasible and cost-efficient approach to autonomous navigation in structured environments [14,15]. Compared to existing studies [1-15], the integration of these two technologies into a compact, ESP32-powered AGV platform represents an innovative contribution, especially for applications in logistics automation and research-based development.

2 Research method

This paper focuses on developing of an AGV in response to the growing need for autonomous transport solutions, particularly in container terminal operations. The efficient transfer of goods in such environments demands accurate localization and effective path planning, especially when navigating areas with various obstacles. In this study, a cost-effective AGV system is

* Corresponding author: nurvidialaksmi@unesa.ac.id

designed by combining AprilTag-based visual localization with the Ant Colony Optimization (ACO) algorithm, using the ESP32 as the main controller.

AprilTag markers are utilized to provide precise, real-time location tracking via an IP camera, while the ACO algorithm enables adaptive, nature-inspired route planning by simulating the food-searching behavior of ant colonies. The integration of both technologies is intended to enhance the AGV's capability in manoeuvring through simulated container terminal scenarios.

2.1 Hardware components

This research utilizes several essential hardware components in the construction of the AGV prototype, each of which plays a specific role in ensuring functionality, autonomy, and connectivity. The main processing unit is the ESP32 microcontroller, a powerful dual-core processor equipped with integrated Wi-Fi and Bluetooth, making it highly suitable for embedded IoT systems and wireless robotic control. It is responsible for receiving commands, processing sensor input, and transmitting motor control signals in real time [16]. For visual localization, an IP Camera (EZVIZ H1C) is deployed. This camera captures real-time video streams over the RTSP protocol and transmits the feed via a Wi-Fi network to a processing unit. AprilTag markers embedded in the environment are detected from this video feed, allowing the AGV to determine its position with visual cues. The camera's high-resolution capability and reliable connection are critical for robust AprilTag detection under varying environmental lighting. The RC Truck Platform serves as the mobile base for the AGV system. This off-the-shelf remote-controlled vehicle is modified by embedding necessary electronics such as the ESP32 board, motor drivers, voltage regulators, and communication modules. Its wheels and steering mechanisms are directly controlled by the microcontroller, allowing autonomous navigation based on visual input hazards. Lastly, the AGV uses multiple AprilTag markers strategically placed within the environment. These markers act as fiducial landmarks used by the vision system to detect and localize the AGV's position. By decoding the marker ID and interpreting its location in the video frame, the system calculates both distance and direction relative to the AGV, which is essential for path planning and navigation.

Each component is carefully selected to balance cost, performance, and integration simplicity, making the AGV prototype a practical platform for research and educational development in the field of autonomous systems.

2.2 Distance estimation

To determine the relative position of an AprilTag with respect to the camera, the pinhole camera model is used, which serves as a fundamental principle in computer vision systems. In this model, the relationship between the real-world object size, the object's size in the image, the camera's focal length, and the distance from the camera can be expressed mathematically [6].

$$d = \frac{f.h}{H} \tag{1}$$

Where d represents the distance between the camera and the AprilTag in meters or centimeters, f is the focal length of the camera in pixels, h is the actual size (height or width) of the AprilTag in meters or centimeters, and h is the apparent size (height or width) of the AprilTag in the image measured in pixels which allows the AGV system to determine its relative position to the marker in real time thereby enhancing navigation accuracy and autonomous path tracking capabilities

The pinhole camera model assumes a simple geometric projection from 3D world coordinates to 2D image coordinates, where the triangle formed by the tag size and distance is similar to the triangle formed by the projection and focal length. The equation (1) allows us to determine the real-world distance of the tag if the camera has been previously calibrated to obtain its focal length.

In a practical implementation, f is obtained from camera calibration using a checkerboard pattern and OpenCV functions, particularly cv2.calibrateCamera(). The size, h is measured directly from the detected tag's bounding box in the frame using AprilTag or ArUco libraries.

2.3 Estimation of horizontal deviation angle

In an image-based navigation system, detecting the horizontal position of an AprilTag relative to the center of the camera frame is crucial for determining the steering direction of the AGV. This positional offset can be mathematically expressed as an angular deviation from the optical axis of the camera. Assuming a pinhole camera model, the angle θ\thetaθ between the center of the image and the center of the detected tag is calculated using the following equation [16]:

$$\theta = arctan\left(\frac{xc - x0}{f}\right) \tag{2}$$

Is used where θ is the horizontal angle of deviation in radians, xc represents the horizontal pixel coordinate of the center of the detected AprilTag, x0 is the horizontal pixel coordinate of the image center, and f is the camera's focal length in pixels which is typically obtained through a prior calibration process enabling the AGV to accurately estimate the angular displacement and adjust its orientation accordingly during autonomous navigation.

Eq. 2 allows the system to determine whether the AprilTag is located to the left or right of the camera center, and by how much. It plays an essential role in adjusting the movement direction of the AGV, especially in systems where the camera serves as the primary sensor for localization.

2.4 Flowchart AGV system

In Figure 1, the flowchart of the AGV system developed by the author is shown. The following is a detailed explanation of each stage in the flowchart.

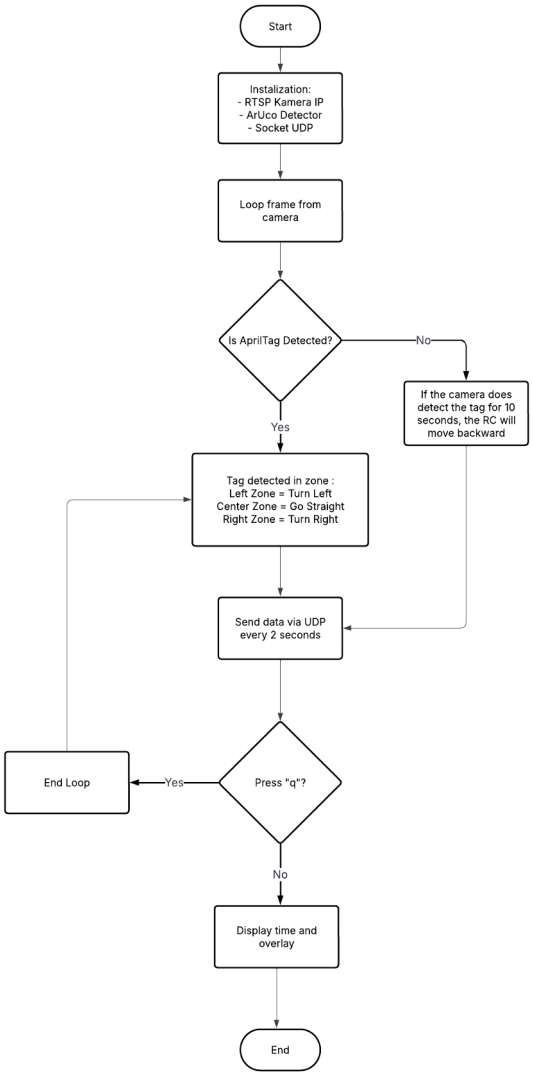

Fig. 1. Flowchart of the AGV system.

The program begins by powering on the computer or image processing device, including the network connection and the IP camera (EZVIZ H1C), followed by an attempt to establish a connection to the RTSP stream from the IP camera, where if the connection fails the system will display an error message and terminate; once connected, the AprilTag detector is initialized using the DICT_APRILTAG_36H11 dictionary, and frames are continuously captured from the video stream for marker detection, where if a marker with a specific target ID (e.g., ID 32) is found, its position is determined along the X-axis to classify it into one of three zones—left, center, or right—with corresponding actions assigned as “Turn Left,” “Go Straight,” or “Turn Right” respectively, and if the marker is not detected for 10 consecutive seconds the system sends a “Reverse” command (code: 2) once before waiting for further detection, while all detected actions (coded as 0, 1, 2, 3, or 4) are transmitted every 2 seconds to the ESP32 via UDP protocol and overlaid in real-time on the video feed for monitoring, and optionally the system can be manually terminated by pressing the "q" key.

2.5 IoT and AI

This research integrates Internet of Things (IoT) and Artificial Intelligence (AI) principles within the AGV prototype. The IoT aspect is implemented through the ESP32 microcontroller, which connects wirelessly to the local network, enabling real-time communication between the vision processing system and the AGV. The ESP32 receives positioning data from the AprilTag detection system and transmits control commands to the vehicle.

For the AI component, a bio-inspired optimization technique known as Ant Colony Optimization (ACO) is implemented. The ACO algorithm simulates ant foraging behavior to compute the most efficient navigation path for the AGV while avoiding obstacles. This integration of IoT and AI allows the AGV to autonomously navigate toward target markers with improved path efficiency and system flexibility. The Ant Colony Optimization (ACO) algorithm is inspired by the behavior of real ants in finding the shortest path from their nest to a food source. Each artificial ant explores the graph and selects a path based on a probability that is influenced by pheromone intensity and heuristic distance. After reaching the destination, the ant returns to the nest while laying down pheromones on the traversed path, reinforcing that route for the next iterations. This process is illustrated in Figure 2.

Ant Colony Optimization (ACO) is a metaheuristic algorithm inspired by the natural foraging behavior of ants, commonly used to solve complex combinatorial optimization problems such as finding the shortest path. In ACO, artificial ants move through a graph to explore possible routes based on two main factors: pheromone trails (T_{ij}), which reflect the desirability of a path based on previous experiences, and visibility $(n_{ij}) = 1/(d_{ij})$, which is the inverse of distance and encourages choosing shorter paths1 [17].

The probability of an ant k moving from node x to node y is given by:

$$P_{xy}^{k} = \frac{(Txy)^{\alpha} \cdot (\eta xy)^{\beta}}{E_{z \in N(x)} (Txy)^{\alpha} \cdot (\eta xy)^{\beta}} \quad (3)$$

Where Txy represents the pheromone intensity on edge xy, and η_{xy} is the heuristic value typically defined as the inverse of the distance between nodes x and y. The parameters α and β control the relative influence of the pheromone trail and heuristic information, respectively. The denominator sums over all feasible neighboring nodes z ∈ N(x)z, which have not yet been visited by the ant, ensuring probabilistic path selection based on both accumulated pheromone and immediate attractiveness. This probabilistic rule enables the colony to balance exploration and exploitation in the search for the shortest or most efficient path [17].

After each cycle, the pheromone levels on paths are updated to reinforce shorter or better routes. The global pheromone update is typically performed using [17]:

$$\tau xy \leftarrow (1-\rho)\tau xy + k = 1\sum m\Delta\tau xyk \quad (4)$$

Rate $(0 < \rho < 1)$, is the number of ants, and $\Delta\tau_{xy}^{k}$ is the amount of pheromone deposited by ant k, often based on the quality (length) of the solution.

After each cycle, the pheromone levels on paths are updated to reinforce shorter or better routes. The global pheromone update is typically performed using:

Figure 3 illustrates the system architecture of the proposed AGV prototype which integrates Artificial Intelligence (AI) through the Ant Colony Optimization (ACO) algorithm and Internet of Things (IoT) technology using wireless communication protocols.

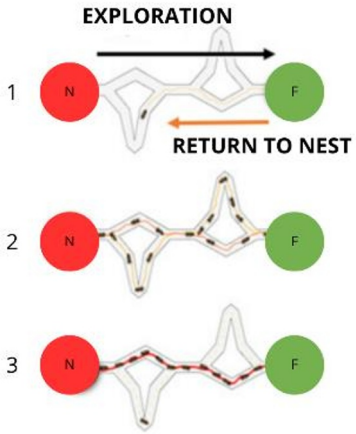

Fig. 2. Process ant colony optimization.

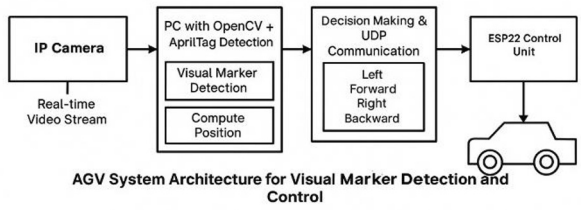

Fig. 3. System architecture.

The system begins with an IP Camera that captures a real-time video stream of the environment. This video feed is processed by a host computer running OpenCV and AprilTag detection algorithms, where visual markers (AprilTags) are identified and their positions are computed. The decision logic, powered by the ACO algorithm, determines the optimal direction (left, right, forward, or backward) based on the detected marker's position and the predefined path optimization criteria. This decision is transmitted via UDP (User Datagram Protocol)—a lightweight communication protocol commonly used in IoT systems—to an ESP32 control unit embedded in the AGV. The ESP32 then interprets the command and actuates the movement of the vehicle accordingly.

2.6 System architecture and workflow

The system architecture implemented in this research is structured into three main functional stages: Input, Processing, and Output, as illustrated in Figure 4. The input stage involves the acquisition of visual information from an IP camera, which is positioned to monitor the environment and detect AprilTag markers strategically placed to act as navigation references. The camera captures frames in real-time and transmits them over a Wi-Fi network to the processing unit. Each detected marker is analyzed to determine its relative position—categorized as left, center, or right—based on its pixel location in the captured image.visual detection and intelligent path planning within a compact, IoT-based prototype platform.

In the processing stage, the ESP32 microcontroller functions as the system's core computational unit. It receives positional data from the vision system and employs the Ant Colony Optimization (ACO) algorithm to compute the most efficient path toward the target. The ACO algorithm simulates the foraging behavior of ants by optimizing route selection based on virtual pheromone trails, enabling the AGV to dynamically adjust its trajectory depending on marker location and path history.

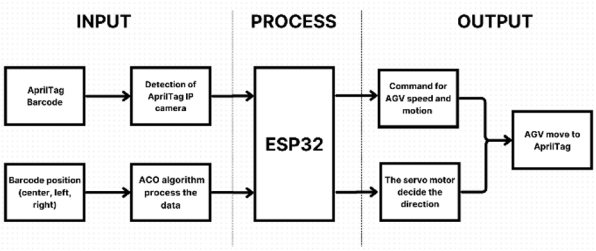

Fig. 4. Block diagram of the AGV system.

The output stage involves actuation and movement execution. Based on the result of the path planning, the ESP32 transmits motion control commands—such as turning left, moving forward, or turning right—to the AGV's motors via pulse-width modulation (PWM) signals. Servo motors are used to accurately steer the vehicle according to the designated direction. This closed-loop interaction ensures that the AGV can respond rapidly to environmental input, maintaining precise alignment with the intended path. By combining visual detection through AprilTag, intelligent decision-making using ACO, and real-time actuation via ESP32, the system forms a compact, IoT-based autonomous navigation platform suitable for research and educational applications. This integrated workflow provides a balance between simplicity and intelligence, supporting low-cost development while maintaining high adaptability in structured environments such as container terminals or indoor labs.

3 Result and discussion

Table 1 shows the results of AprilTag detection accuracy tests. In each trial, ten AprilTags were placed in the AGV's environment, and the system attempted to detect them using the IP camera and AprilTag

algorithm. The average detection accuracy achieved was 96.6%, demonstrating the system's reliability under standard lighting conditions.

Table 1. AprilTag detection accuracy.

Trial	TotalTag Tested	Detected	A (%)
1	10	10	100%
2	10	9	90%
3	10	10	100%
Average	—	—	96.6%

$$Accuracy = \frac{Number\ of\ Tags\ Detected}{Total\ Number\ of\ Tags} \times 100\% \quad (5)$$

$$Accuracy = \frac{29}{30} \times 100\% = 96.6\ \% \quad (6)$$

This level of accuracy is sufficient for real-time navigation purposes, as it minimizes the risk of missed detections that could affect AGV path planning. The prototype AGV successfully detected AprilTags within the environment, maintaining a localization accuracy of over 96,6% under standard lighting conditions. Real-time position data allowed for continuous adjustment of the AGV's trajectory.The ACO algorithm generated efficient navigation paths, consistently reducing total travel distance by approximately 15% compared to baseline manual routes. The system exhibited low latency in processing visual input and executing movement commands, demonstrating the viability of combining AprilTag detection and ACO for real-time autonomous navigation.

Figure 5 depicts the main hardware components assembled in the AGV prototype system. On the left side, the image shows a modified RC truck which functions as the mobile platform for the AGV. Placed on top of this vehicle is an IP camera that captures live video feed and performs AprilTag detection to aid in localization. The central part of the figure shows the power supply system, composed of 18650 batteries and a TP4056 charging module, which delivers consistent electrical power for portable operation. In the bottom section, the ESP32 microcontroller is presented as the core controller responsible for processing incoming data, managing wireless communication, and issuing navigation commands. Meanwhile, the image on the right shows the remote-control module, which is adapted to communicate with the ESP32 and control the movement and direction of the RC vehicle. Altogether, these hardware components enable the AGV to operate autonomously by combining real-time visual input with intelligent motion control.

Figure 6 shows the electrical schematic of the AGV prototype developed in this study. The system uses a 18650 lithium battery as the main power source, which is regulated by the TP4056 power module to ensure stable voltage output of 5V. The TP4056 module also provides safe battery charging capability without needing to replace the battery.

The ESP32 microcontroller serves as the central controller, connected to various components through a breadboard for ease of prototyping and system testing. Push buttons and LEDs are added to facilitate manual control and visual indicators during the development phase.

This modular configuration provides flexibility in testing, modification, and troubleshooting, which is essential during the prototyping stage. The compact layout also allows the system to be mounted onto the RC platform, forming a functional AGV prototype capable of autonomous navigation.

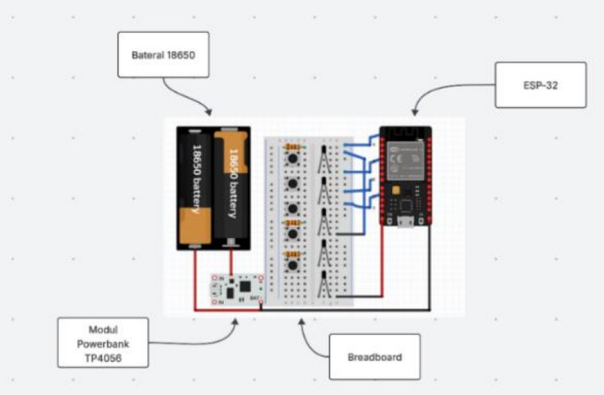

Fig. 5. AGV system hardware.

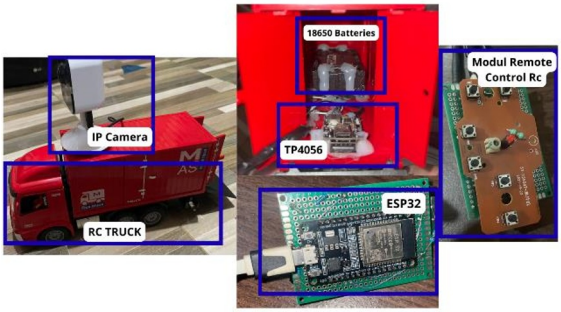

Fig. 6. Electrical schematic.

3.1 Testing of AprilTag detection

Figure 7 shows the real-time detection interface of the AprilTag marker during system testing. The video stream is captured using the IP camera, processed by the AprilTag detection algorithm, and displayed on a computer screen with relevant information overlays.

In this trial, the marker with ID 32 is successfully detected, as indicated by the bounding box and marker information displayed on the screen. The detected position of the marker is used as a reference for the AGV's navigation commands.

The selection of ID 32 is both strategic and technical, as this ID features a unique pattern that is neither too simple nor too symmetrical, reducing the risk of false detection and making it ideal for control and navigation experiments; using a consistent ID allows the system to focus on identifying a single target, which is essential for evaluating the performance of the navigation and detection algorithms in a stable manner; moreover, ID 32 is part of the tag36h11 dictionary and does not conflict with other tags used in the testing environment, preventing duplicate detections or decision-making errors.

The system also displays the current action ("action") based on the marker's position relative to the defined zones (left, center, right). In this case, the marker is located in the center zone, prompting the AGV to move forward automatically.

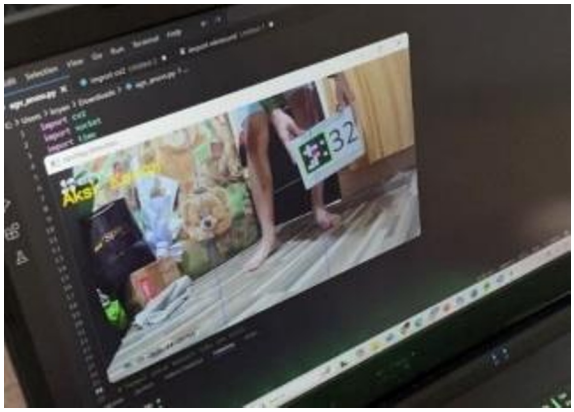

Fig. 7. Real-time AprilTag detection interface.

4 Conclusions

Based on the development and testing conducted in this research, it can be concluded that the ESP32-based AGV prototype integrated with AprilTag detection and the Ant Colony Optimization (ACO) algorithm functions effectively for autonomous navigation in a controlled environment.The AprilTag system successfully provided real-time position detection with high accuracy, enabling the AGV to recognize visual markers and adjust its trajectory accordingly. Furthermore, the implementation of the ACO algorithm optimized the AGV's path by determining the most efficient route toward the target marker, while considering environmental obstacles. The integration of IoT-based microcontrollers (ESP32), computer vision (AprilTag), and bio-inspired optimization algorithms (ACO) has proven to be a low-cost yet reliable solution for AGV navigation systems. The prototype demonstrates significant potential for applications in container terminals and other industrial environments requiring autonomous transport solutions. Future improvements, such as the addition of obstacle detection sensors (e.g., ultrasonic or LIDAR) and enhanced visual systems, are recommended to increase the system's robustness in more dynamic and complex environments.

The author would like to express sincere gratitude to Universitas Negeri Surabaya and CV. Denis Sukses Mandiri for providing the facilities, technical support, and opportunity to complete this research project. Special thanks are extended to Mrs. Nur Vidia Laksmi B., S.ST., M.Sc., as the academic supervisor, and Mr. Nyoman Yudi Kurniawan, S.T., for the valuable guidance and mentorship throughout the prototype development process. The author also appreciates the assistance and support from family and colleagues, which contributed to the success of this research.

References

1. C.B. Asaju, P.A. Owolawi, C. Du, E. Van Wyk, Enhancing Security with Automated Boom Gate Access Through License Plate Recognition Utilising yolov8 Model, in: Stahlbock, R., Arabnia, H.R. (eds) Data Science. CSCE 2024. Communications in Computer and Information Science, 2253. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-85856-7_15
2. Bhargava, A., Suhaib, M., & Singholi, A. S., A review of recent advances, techniques, and control algorithms for automated guided vehicle systems. Journal of the Brazilian Society of Mechanical Sciences and Engineering, **46**, 7, 419 (2024). https://doi.org/10.1007/s40430-024-04896-w
3. Tsolakis, N., Zissis, D., Papaefthimiou, S., & Korfiatis, N., Towards AI driven environmental sustainability: an application of automated logistics in container port terminals. International Journal of Production Research, **60**, 14, 4508-4528 (2022). https://doi.org/10.1080/00207543.2021.1914355
4. Palos, M., Cortés, I., Madridano, Á., Navas, F., Barbero, C., Milanés, V., & García, F., LiDAR-based perception system for logistics in industrial environments. Applied Intelligence, **55**, 7, 1-20. (2025). https://doi.org/10.1007/s10489-025-06528-9
5. He, L., Liu, S., & Shen, Z. J. M., Smart urban transport and logistics: A business analytics perspective. Production and Operations Management, **31**, 10, 3771-3787 (2022). https://doi.org/10.1111/poms.13775
6. Košťák, M., & Slabý, A., Designing a simple fiducial marker for localization in spatial scenes using neural networks. Sensors, **21**, 16, 5407 (2021). https://doi.org/10.3390/s21165407
7. A. Umar, I. Abdulrazaq Imam, A. O. O. Ore-Ofe, D. Daniel Emmanuel, D. Sylvester Aondonenge, and L. Abdulwahab Olugbenga, Assessing the Strategic Impact of Artificial Intelligence - Robotic Process Automation on Enterprise Architecture in the Telecommunications Industry. Vokasi Unesa Bull. Eng. Technol. Appl. Sci. **1**, 3, 28–40 (2024). https://doi.org/10.26740/vubeta.v1i3.36736
8. A. Hildayanti and M. S. Machrizzandi, Image optimization technique using local binary pattern and multilayer perceptron classification to identify potassium deficiency in cacao plants through leaf images. Vokasi Unesa Bull. Eng. Technol. Appl. Sci. **2**, 1, 77–87 (2025). https://doi.org/10.26740/vubeta.v2i1.34587
9. H. Zangana, M. M. Firas, and M. Omar, A novel hybrid algorithm for effective image restoration. Vokasi Unesa Bull. Eng. Technol. Appl. Sci. **2**, 2, 175–188 (2025). https://doi.org/10.26740/vubeta.v2i2.38118
10. Vlachos, I., Pascazzi, R. M., Ntotis, M., Spanaki, K., Despoudi, S., & Repoussis, P., Smart and flexible manufacturing systems using Autonomous Guided Vehicles (AGVs) and the Internet of Things (IoT). International Journal of Production Research,

62(15), 5574-5595 (2024). https://doi.org/10.1080/00207543.2022.2136282
11. Oliveira, J., Fernandes, B., & Henriques, M, Optimizing Internal Logistics Using Automated Guided Vehicles: An Evaluation of Heuristic Approaches. International Conference on Innovative Intelligent Industrial Production and Logistics. pp. 42-55. Cham: Springer Nature Switzerland (2024). https://doi.org/10.1007/978-3-031-80775-6_3
12. Priyadarshi, R., & Kumar, R. R., Evolution of swarm intelligence: a systematic review of particle swarm and ant colony optimization approaches in modern research. Archives of Computational Methods in Engineering, 1-42 (2025). https://doi.org/10.1007/s11831-025-10247-2
13. Alqudsi, Y., & Makaraci, M., Exploring advancements and emerging trends in robotic swarm coordination and control of swarm flying robots: A review. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **239**, 1, 180-204 (2025). https://doi.org/10.1177/09544062241275359
14. C. C. Lai, B.J Yang, C. J Lin, Applying Reinforcement Learning for AMR's Docking and Obstacle Avoidance Behavior Control, Applied Sciences (Switzerland). **15**. 7 (2025). https://doi.org/10.3390/app15073773
15. Kubasakova, I., Kubanova, J., Benco, D., & Kadlecová, D., Implementation of automated guided vehicles for the automation of selected processes and elimination of collisions between handling equipment and humans in the warehouse. Sensors, **24**. 3. 1029 (2024). https://doi.org/10.3390/s24031029
16. Yang, Y., & Pan, W., Automated guided vehicles in modular integrated construction: Potentials and future directions. Construction Innovation, **21**, 1, 85-104 (2021). https://doi.org/10.1108/CI-07-2019-0073
17. Abbas, S. M., Aslam, S., Berns, K., & Muhammad, A., Analysis and improvements in AprilTag based state estimation. Sensors, **19**, 24, 5480 (2019). https://doi.org/10.3390/s19245480