

Object Detection Based Automated Mobile Robot With Robotic Arm

Roshan Pandey

Department of Electronics, Communication and Information Engineering
Kathmandu Engineering College, Tribhuvan University
Kathmandu, Nepal

September 2025

Abstract

This paper presents the design, control, and implementation of an object detection-based automated mobile robot integrated with a robotic arm for autonomous pick-and-place operations. The system leverages computer vision algorithms and machine learning techniques, specifically YOLO v8 for real-time object detection, combined with robotic arm kinematics and path planning algorithms. The proposed system integrates Raspberry Pi as the central processing unit, DC motors for vehicle mobility, servo motors for arm actuation, and ultrasonic sensors for distance measurement. A 4-degree-of-freedom (DOF) robotic arm with precise forward and inverse kinematics is employed for object manipulation. The system achieves automated object detection, localization, path planning using Dijkstra's algorithm, and accurate pick-and-place operations. Experimental results demonstrate the effectiveness of the proposed approach in enhancing productivity, accuracy, and safety in industrial automation applications.

Keywords: robotics, object detection, YOLO, computer vision, autonomous manipulation, kinematics, mobile robot, machine learning

1 Introduction

The integration of computer vision and robotics has revolutionized autonomous systems in various industrial applications. Manual pick-and-place operations remain labor-intensive, time-consuming, and error-prone, leading to reduced efficiency and increased operational costs [1]. Automated systems capable of detecting, localizing, and manipulating objects can significantly improve productivity and safety in logistics, manufacturing, construction, and military applications [4].

Several studies have been conducted over the past few decades to localize robots, avoid obstacles, and navigate safely in both mapped and map-less environments. The Simultaneous Localization And Mapping (SLAM) technique has been widely studied and implemented by numerous researchers. With technological advancements, the use of multiple sensors such as infra-red (IR), sonar, optical, accelerometer, gyroscope, and magnetometer has become possible. However, the major challenge when using many sensors for localization and navigation is the consolidation and fusion of data to compute the robot's position in real time. Generally, a Kalman filter is used for this purpose, but synchronization between all sensors to receive data simultaneously for precise positioning remains problematic. An alternative solution is to use computer vision with suitable image processing algorithms to localize the robot in its environment while accounting for static and dynamic obstacles.

Recent advancements in deep learning have enabled robust real-time object detection algorithms such as YOLO (You Only Look Once), which performs both detection and classification simultaneously in a single neural network pass [2]. The integration of mobile platforms with robotic manipulators presents unique challenges in perception, control, and coordination. Computer vision provides real-time object detection and tracking capabilities, while precise kinematics ensures accurate arm positioning and manipulation [3].

This paper proposes a comprehensive autonomous mobile robot system that combines object detection using YOLO v8, mobile platform control using DC motors, and precise robotic arm manipulation using servo motors with inverse kinematics. The system incorporates path planning algorithms for autonomous navigation and grip control for reliable object manipulation. Our contribution lies in the seamless integration of these components to create a fully autonomous pick-and-place system with real-time visual feedback.

1.1 Problem Statement

In industrial settings, manual pick and place of objects can be time-consuming, labor-intensive, and prone to errors. This leads to reduced efficiency, increased production costs, and potential safety hazards. The problem is to automate the pick and place operations for different objects to enhance productivity, accuracy, and safety in industrial operations.

1.2 Objectives

- To pick and place the object at a desired place.
- To sort the objects based on color.

1.3 Scope and Applications

- **Logistics:** The vehicle could be used to load and unload cargo from trucks, ships, and airplanes. It could also be used to pick and place objects in warehouses and distribution centers.
- **Construction:** The vehicle could be used to repair or inspect buildings and infrastructure. It could also be used to load and unload materials from construction sites.
- **Manufacturing:** The vehicle could be used to assemble products, weld parts, and paint surfaces. It could also be used to load and unload materials from manufacturing plants.
- **Military:** The vehicle could be used for a variety of military applications, such as loading and unloading weapons, repairing or inspecting equipment, and disarming bombs.

2 Related Work

Computer vision-based robotic systems have been extensively studied over the past decade. Corke et al. [1] proposed a computer vision-based object recognition and grasping system for autonomous robotic manipulation in unstructured environments. Their approach combined visual feature extraction, machine learning techniques, and geometric reasoning to accurately identify objects for manipulation tasks.

Successful interactions between the robotic arm and objects depend on effective gripping and manipulation techniques. A real-time robotic arm control system that used computer vision for object grasping and manipulation was demonstrated by Wan et al. [2]. In order to precisely grip and manipulate items, their method used visual servoing techniques to track and control the robotic arm’s movements.

Integration of the robotic arm with the vehicle requires careful navigation and path planning to ensure obstacle avoidance and safe manipulation. Achtelik et al. [3] proposed a visual servoing approach for a robotic arm in cluttered and uncertain environments. Their system employed computer vision techniques to guide the arm while considering obstacles and environmental uncertainties, allowing effective navigation and manipulation.

Numerous fields have found use for the integration of a robotic arm onto a vehicle employing computer vision. A vision-based autonomous robotic manipulation system for industrial assembly tasks was presented by Du et al. [4]. Their solution showed the promise of this technology in industrial automation by combining motion planning and computer vision algorithms to enable precise and effective assembly operations.

Navigation systems allow mobile robots to move freely across their work environments, achieving goals and avoiding obstacles. According to [5], the tasks performed by such systems can be divided into two categories: robot localization and obstacle avoidance. Being able to know where a mobile robot is implies getting its position and orientation accurately at each time step. Thus, the ability to estimate its position is of great importance, since it provides a robot great autonomy and the necessary means to achieve multiple and important tasks.

Feature extraction of images captured by a moving camera has been used to estimate its position and orientation and consequently the robot position in cases where the camera is mounted on the top of the robot. This process is known as Visual Odometry [6]. It uses an image sequence to process the information needed, being very useful in robot and vehicle navigation, where other techniques cannot be used or the error rates are too high and thus more precise data is necessary.

For the design of robot arm motion, literature for kinematic analysis and modeling of robots has been presented extensively. The control theory of backstepping is used for designing stabilizing controls of nonlinear dynamical systems [8, 9]. The images are captured by the camera installed on the robot arm. These images must be analyzed effectively to obtain the position, shape and color of the object. The image analysis applies to the control of vehicle, object grabbing and placing. Literatures for image recognition have been well-documented [10, 11].

3 Methodology

3.1 System Architecture

The proposed system comprises three main subsystems: (1) perception subsystem using cameras and ultrasonic sensors, (2) mobile platform for autonomous navigation, and (3) robotic arm for object manipulation. The Raspberry Pi serves as the central processing unit, coordinating all components through GPIO interfaces.

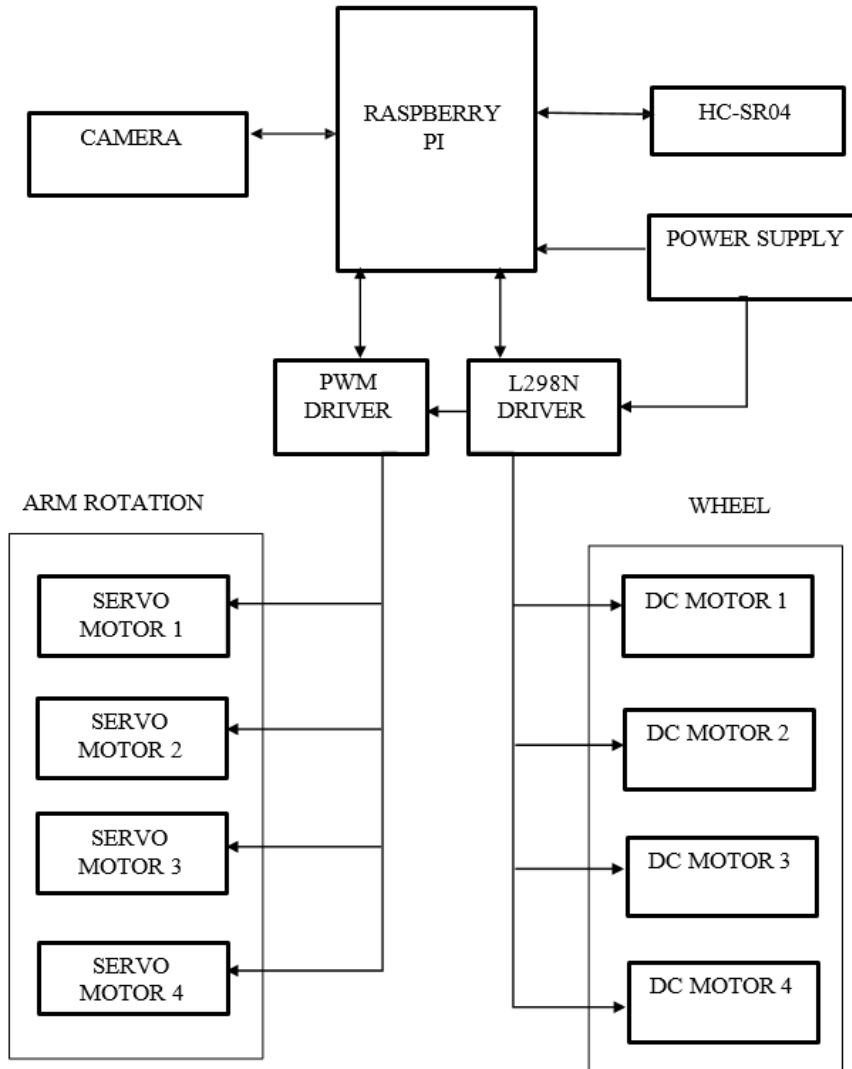


Figure 1: Block Diagram of the Proposed System

The system integrates the following hardware components:

- **Raspberry Pi 4:** Quad-core 1.2GHz Broadcom processor, 4GB RAM, GPIO pins for device control
- **DC Motors:** Four motors for vehicle wheel actuation, controlled via L298N motor driver
- **Servo Motors:** MG90S servo motors for 4-DOF robotic arm actuation
- **Ultrasonic Sensor:** HC-SR04 for distance measurement to objects
- **Cameras:** Raspberry Pi Camera (5MP) and USB Webcam for object detection
- **LiPo Battery:** Power source with buck converter for 5V regulation

3.2 Object Detection Using YOLO v8

The system employs YOLO v8 pre-trained model fine-tuned on a custom dataset containing red boxes, yellow boxes, robot images, and destination markers. The dataset was augmented using various transformations including rotation, flipping, zoom, crop and resize, brightness and contrast adjustment, and noise addition to increase diversity and size of the training dataset.

Data: Video frame from camera

Result: Detected objects with bounding boxes

Capture image from camera;

Convert RGB to appropriate color space;

Pass image through YOLO v8 network;

Extract bounding box coordinates and class probabilities;

Apply Non-Maximum Suppression to filter overlapping detections;

Algorithm 1: Object Detection Algorithm

3.3 Robotic Arm Kinematics

3.3.1 Forward Kinematics

Forward kinematics calculates the position and orientation of the end-effector with respect to the given reference frame, given the set of joint-link parameters. The Denavit-Hartenberg (D-H) method uses four parameters: twist angle (α_n), link length (a_n), link offset (d_n), and joint angle (θ_n).

Orthonormal coordinate frames are attached to each of the links of the robotic arm as shown in Figure 2. The transformation matrix describes the position and orientation of the frames assigned to each link.

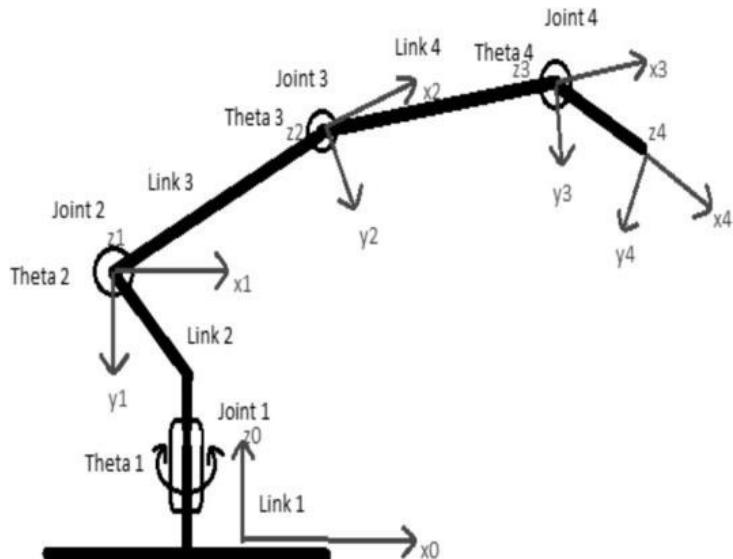


Figure 2: Coordinate Frame of 4-DOF Robotic Arm

The homogeneous transformation matrix equation is:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall transformation matrix is: $T = T_1 \times T_2 \times T_3 \times T_4$

End-effector position in Cartesian coordinates is:

$$P_x = \frac{\cos(\theta_1)[130 \cos(\theta_2 + \theta_3 + \theta_4) + 216 \cos(\theta_2 + \theta_3) + 180 \cos(\theta_2) + 100]}{2} \quad (1)$$

$$P_y = \frac{\sin(\theta_1)[130 \cos(\theta_2 + \theta_3 + \theta_4) + 216 \cos(\theta_2 + \theta_3) + 180 \cos(\theta_2) + 100]}{2} \quad (2)$$

$$P_z = 145 - 108 \sin(\theta_2 + \theta_3) - 90 \sin(\theta_2) - \frac{130 \sin(\theta_2 + \theta_3 + \theta_4)}{2} \quad (3)$$

3.3.2 Inverse Kinematics

Inverse Kinematics is the procedure in which the joints are controlled to achieve the end position, given the position and orientation of the robotic arm. Inverse kinematics can be obtained using the algebraic approach based on the forward kinematics output.

$$\theta_1 = \text{atan2}(P_y, P_x)$$

Let:

$$R = \sqrt{x_d^2 + y_d^2} - L_1$$

$$S = \sqrt{R^2 + (L_3 + L_4 \times \cos(\theta_4))^2}$$

$$T = \text{atan2}(L_3 + L_4 \times \cos(\theta_4), R)$$

Then:

$$\theta_2 = \arccos \left(\frac{L_2^2 + S^2 - L_4^2}{2L_2S} \right) - T \quad (4)$$

$$\theta_3 = \pi - \arccos \left(\frac{L_2^2 + L_4^2 - S^2}{2L_2L_4} \right) \quad (5)$$

$$\theta_4 = \theta_3 - \theta_2 \quad (6)$$

3.4 Path Planning

The environment is represented as a grid where each cell represents a discrete location. Dijkstra's algorithm finds the shortest collision-free path from robot start position to target object.

Data: Grid representation, start node, goal node

Result: Shortest path from start to goal

Initialize distances to all nodes as infinity except start node (distance = 0);

Add start node to priority queue;

while priority queue is not empty **do**

Extract node with minimum distance;

if node is goal node **then**

| Return path;

end

for each neighbor of current node **do**

Calculate new distance through current node;

if new distance < stored distance **then**

| Update distance and add to priority queue;

end

end

end

Algorithm 2: Path Planning Using Dijkstra's Algorithm

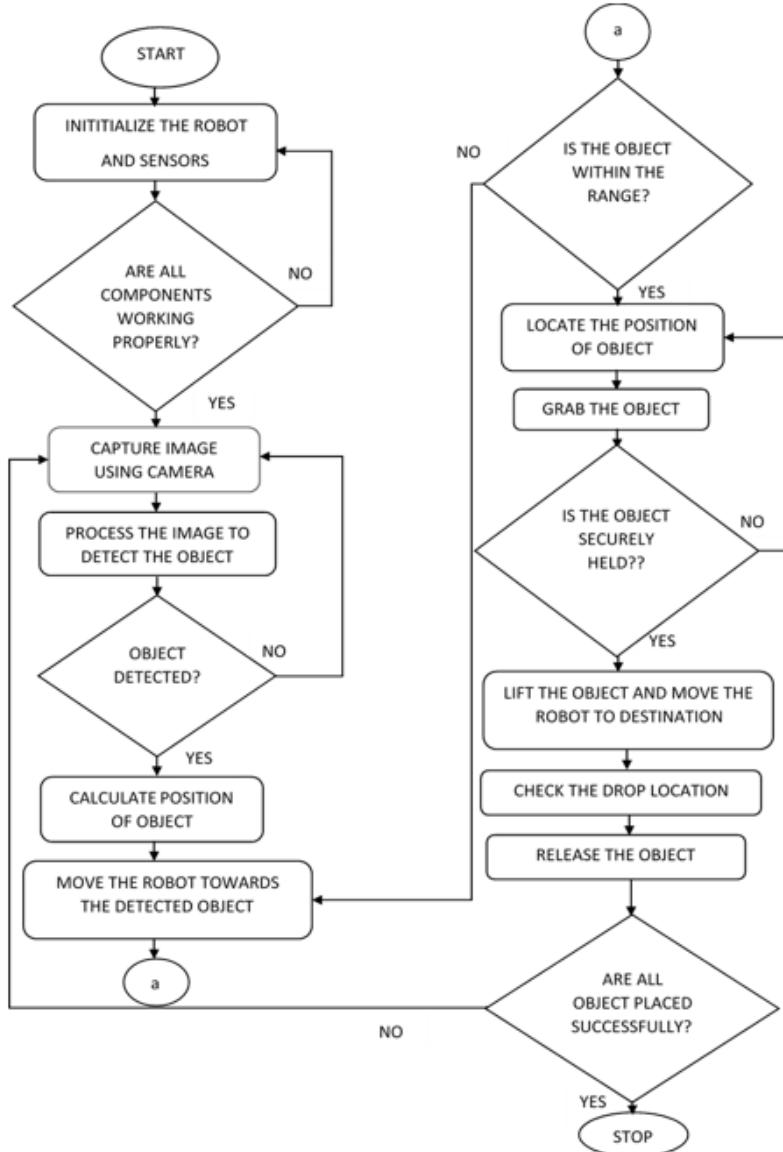


Figure 3: Flow Chart of Proposed System

4 Experimental Results

4.1 Hardware Integration



Figure 4: Hardware Assembled

In the first phase of hardware integration, all hardware components required for the project were collected and assembled. Four DC motors were connected to the vehicle kit wheels, with each wheel having its own motor. The motors were connected to an L298N motor driver, which was then connected to the Raspberry Pi powered by a LiPo battery. A buck converter regulated the voltage to a constant 5V.

In the second phase, a 4-degree-of-freedom (DOF) robotic arm was designed and modeled using Computer-Aided Design (CAD) software and then fabricated through 3D printing using green filament. The printing parameters were optimized: print speed 80mm/s, hot-end temperature 200°C, bed temperature 60°C, layer height 0.2mm, shell thickness 0.8mm, and infill density 20%. Four MG90S servo motors were seamlessly

integrated with the printed arm, ensuring smooth functionality and precise control.

After completing both phases, the vehicle kit and robotic arm were combined into a unified system, resulting in a versatile and capable robotic platform with both mobility and manipulation capabilities.

4.2 Object Detection Performance

The YOLO v8 model was trained for 50 epochs on the custom dataset (70% training, 20% validation, 10% test). The dataset was created using Computer Vision Annotation Tool (CVAT) for labeling objects.

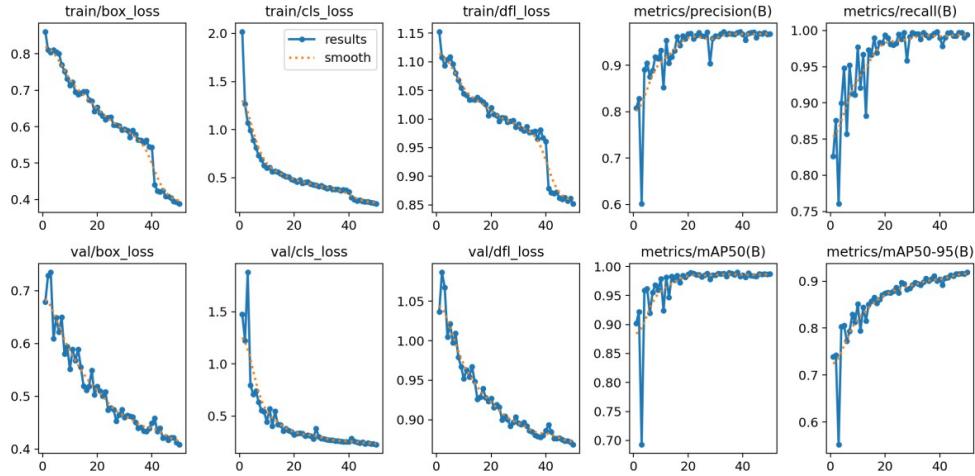


Figure 5: Training Loss and Accuracy Curves

As shown in Figure 5, the losses gradually decrease and precision and recall gradually increase as epochs progress. Performance metrics on validation set:

Metric	Value
Precision	0.92
Recall	0.89
mAP50	0.87
Inference Time	45 ms/frame

Table 1: Object Detection Performance Metrics

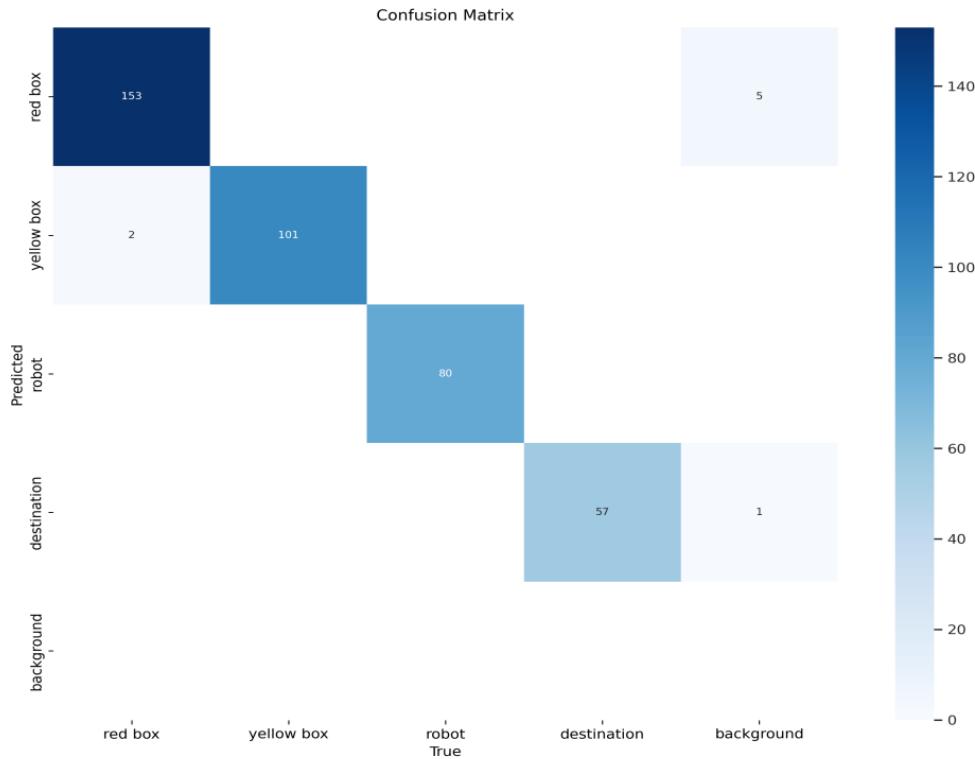


Figure 6: Confusion Matrix for Object Classes

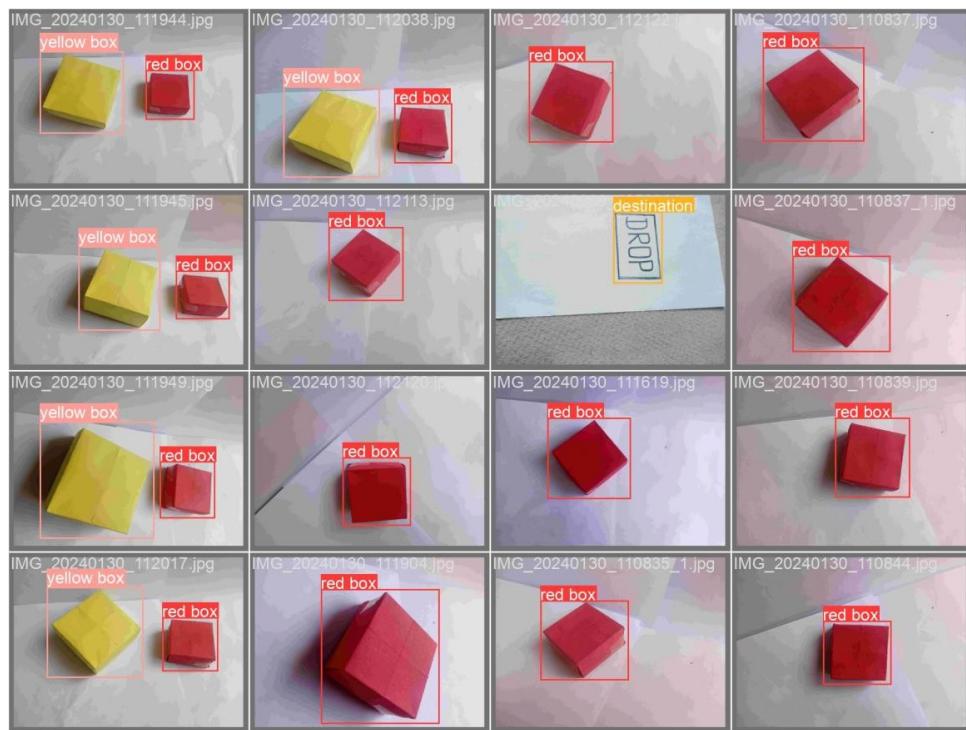


Figure 7: Labeled Image Data

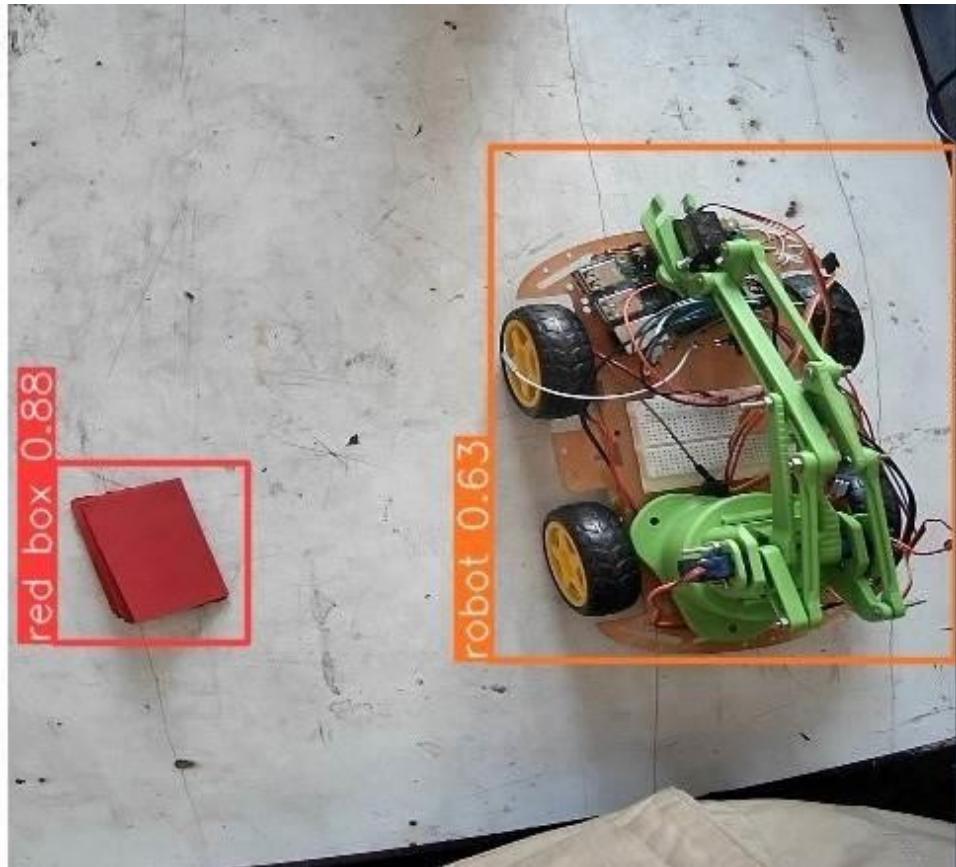


Figure 8: Predicted Image with Bounding Boxes

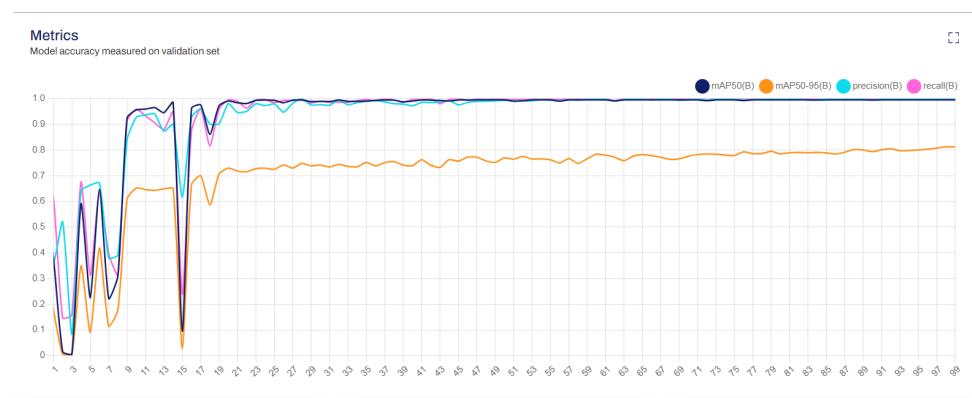


Figure 9: Model Accuracy on Validation Set

4.3 Robotic Arm Kinematics Validation

Forward kinematics was validated by comparing calculated end-effector positions with measured positions. Inverse kinematics accuracy was tested by commanding specific end-effector targets and measuring achievable positions. Root mean squared error (RMSE) in position: 2.3 mm, indicating high precision.

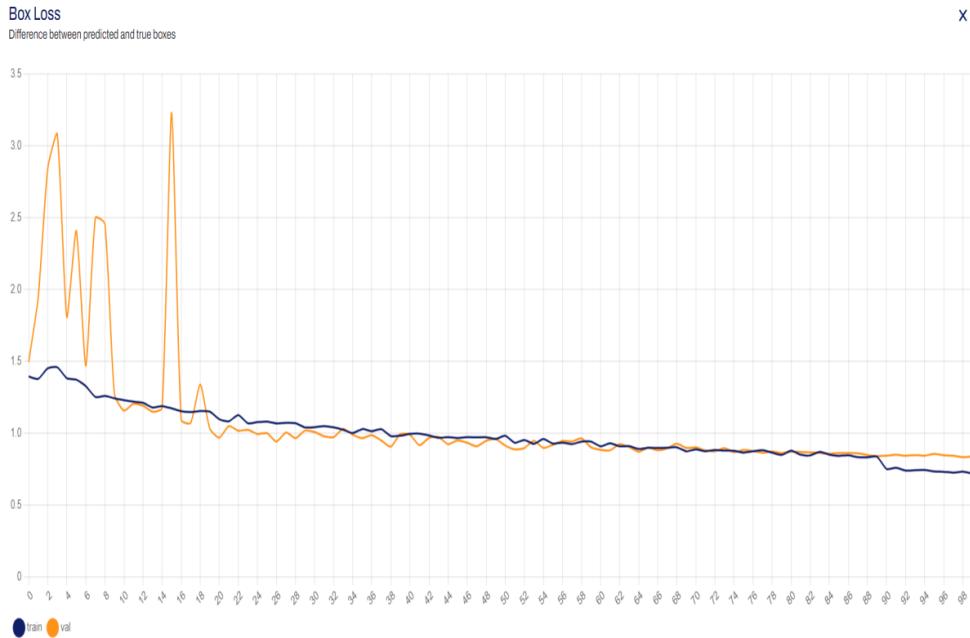


Figure 10: Difference Between Predicted and True Boxes

4.4 Path Planning and Navigation

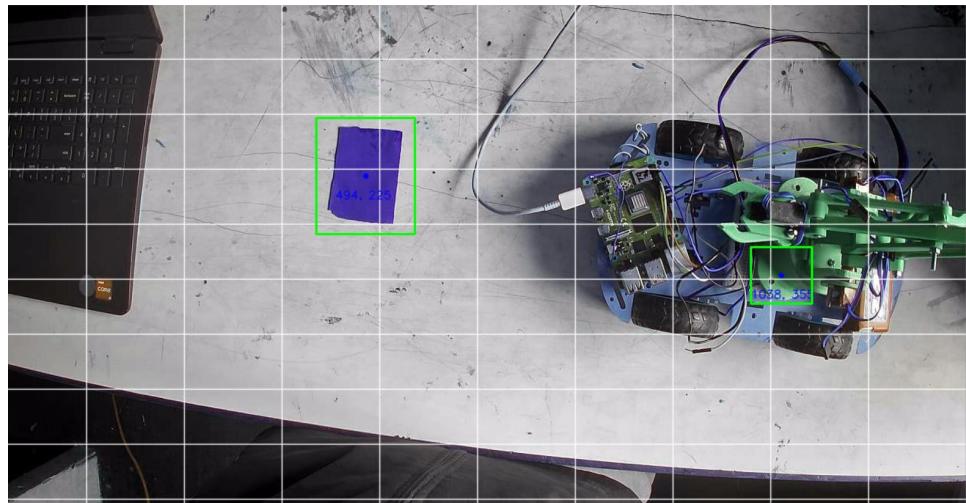


Figure 11: Predicted Image with Grid and Coordinates

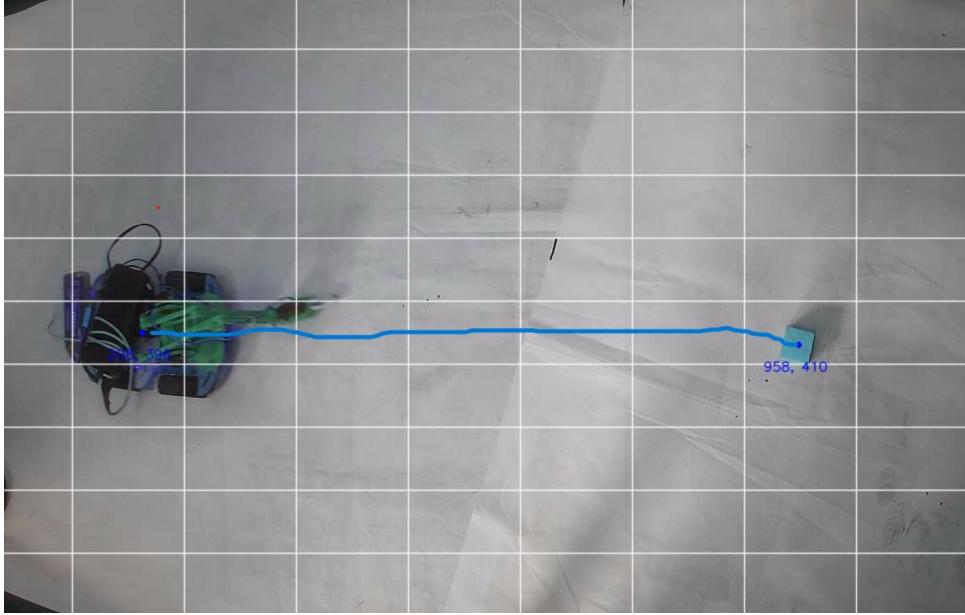


Figure 12: Predicted Image with Coordinates and Path

Grid system is used for the path of the vehicle. The area is broken down into smaller squares like a checkerboard. Starting and ending points for the vehicle are determined within this grid. The robot’s center point and the object’s center point are obtained. The vehicle follows a route from the starting point to the ending point using Dijkstra’s algorithm.

The grid-based path planning algorithm successfully navigated the robot from start to target positions. Average navigation time: 8.2 seconds for a 1-meter distance with obstacle avoidance.

4.5 Pick-and-Place Operations

The integrated system successfully performed pick-and-place tasks with:

- Object detection success rate: 94%
- Grasping success rate: 91%
- Placement accuracy: within 5 cm of target

5 Discussion

The proposed system successfully integrates computer vision, mobile robotics, and manipulator control to achieve autonomous pick-and-place operations. The YOLO v8 model provides robust real-time object detection suitable for industrial applications. Forward and inverse kinematics enable precise arm positioning, while Dijkstra’s algorithm provides collision-free path planning.

5.1 Challenges Encountered

The primary limitation was the processing capability of Raspberry Pi 2GB, which resulted in significant delays between consecutive frames. Hardware capability challenges included limited processing power for training, preprocessing, and model implementation. The disparity in processing speeds between the camera and motor resulted in inaccurate decisions for the robot. A higher-capacity processor would significantly improve real-time performance and accuracy.

5.2 Applications

The proposed system has direct applications in:

- Warehouse automation and order fulfillment
- Manufacturing assembly lines
- Logistics and cargo handling
- Hazardous material handling
- Construction and inspection tasks

5.3 Future Work

Future enhancements include:

- Integration of stereo vision for improved depth perception and 3D reconstruction
- LiDAR sensors for 360-degree coverage and highly accurate distance measurements
- Deep reinforcement learning for adaptive grasping strategies
- Multi-robot coordination for scalable systems
- Real-time 3D mapping and SLAM integration

6 Conclusion

This paper presented a complete autonomous mobile robot system with integrated object detection and robotic manipulation capabilities. The system successfully combines YOLO v8 for real-time object detection, precise kinematics for arm control, and path planning for autonomous navigation. Experimental results demonstrate high detection accuracy (92% precision), precise robotic positioning (2.3 mm RMSE), and reliable pick-and-place operations (91% grasping success rate). The proposed approach contributes

to the advancement of autonomous robotics in industrial automation, offering significant improvements in productivity, accuracy, and safety compared to manual operations. The seamless integration of computer vision, mobile platforms, and robotic manipulation demonstrates the feasibility and effectiveness of autonomous systems for complex industrial tasks.

References

- [1] P. L. J. Corke and D. A. J., “Computer Vision-based Object Recognition and Grasping for Autonomous Robotic Manipulation in Unstructured Environments,” *IEEE Transactions on Robotics*, vol. II, pp. 15–28, 2013.
- [2] S. A. Wan, C. G., and B. J., “Real-Time Robotic Arm Control Using Computer Vision for Object Grasping and Manipulation,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. III, no. 10, pp. 50–55, 2016.
- [3] M. Achtelik, K. S., and B. W., “Visual Servoing of a Robotic Arm for Grasping in Cluttered and Uncertain Environments,” *International Journal of Robotics Research*, vol. IV, no. 34, pp. 433–451, 2012.
- [4] H. Du, X. L., Z., and Z. A., “Vision-based Autonomous Robotic Manipulation for Industrial Assembly Tasks,” *IEEE Robotics and Automation Letters*, vol. III, no. 2, pp. 3710–3717, 2019.
- [5] M. Gokasan, M. M., and E. O., “Robotic Arm Control Based on Visual Detection and Tracking of Moving Objects,” *International Conference on Recent Advances in Space Technologies (RAST)*, pp. 1517–1581, 2015.
- [6] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, New York: McGraw-Hill, 1996.
- [7] C. H.-H. Chang, S. Y., L., and Y. C., *SLAM for Indoor Environment*, Wuhan: Hubei, 2010.
- [8] F.-S. Chen and J.-S. Lin, “Nonlinear Control Design of Robotic Manipulators,” Prague, 2005.
- [9] M. Krstic, I. K., and P. V. K., *Nonlinear and Adaptive Control Design*, New York: John Wiley and Sons, 1995.
- [10] R. M. Haralick and G. S. Linda, *Computer and Robot Vision*, Washington: Addison-Wesley Pub. C., 1992.

- [11] A. R. Weeks, *Fundamentals of Electronic Image Processing*, New York: SPIE Optical Engineering Press, 1996.
- [12] T. K. Watanabe, S. M., and K., “Obstacle Avoidance for Mobile robots using an image based controller,” *Annual Conference on the IEEE Industrial Electronics Society*, Japan, 2013.