Design and Development of a Palletizing Robot with an Intelligent 3D Vision System based on RF-DETR Model

Le Hoai Phuong
Doctor
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Dang Anh Duy
Lecturer
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

This paper presents the design and implementation of a 4-degree-of-freedom (4-DOF) palletizing robotic arm integrated with an intelligent 3D vision system based on RF-DETR. The system targets the challenges of object detection and manipulation in unstructured industrial environments characterized by clutter, occlusion, and object variability. The robot arm features a lightweight, high-stiffness structure driven by AC servo motors and harmonic gearboxes, controlled by a Mitsubishi FX5U PLC. High-level vision processing is performed on an external computer connected via the RS-485 protocol. The RF-DETR model is trained on a custom dataset of common retail and logistics objects, including boxes, bottles, and deformable packages with diverse shapes, colors, and materials. Bounding box outputs are projected into 3D grasp coordinates using depth data from an Intel RealSense camera and converted to robot coordinates through calibrated transformations. Compared with YOLOv12 and LW-DETR, the proposed model achieved superior detection performance (mAP@0.5:0.95 = 0.78), particularly under partial occlusion, where YOLOv12 failed. The system was validated through 2,000 grasping trials across ten object categories, achieving an overall success rate of 98.75%. All rectangular objects were grasped with 100% accuracy, while minor errors occurred with cylindrical items due to slippage. These results demonstrate the proposed system's robustness in complex scenarios and its practical viability for cost-effective, AI-driven palletizing in real-world industrial settings.

Keywords: *RF-DETR; Yolov12, palletizing robot; object detection; 3D vision system.*

1. INTRODUCTION

The rapid advancement of industrial automation in re–cent decades has led to significant transformations across the manufacturing, warehousing, and logistics sectors. Among the various repetitive and labor-inten–sive tasks, palletizing—i.e., the process of stacking pro–ducts or packages onto pallets—remains a critical operation in virtually every production line. Traditi–onally performed by manual labor or pre-programmed industrial manipulators, palletizing has now become a prime candidate for intelligent robotic automation, whe–re flexibility, accuracy, and efficiency are essential. As manufacturing shifts toward smart factories and Indus–try 4.0 paradigms, the need for adaptive and cost-effec–tive palletizing solutions becomes increasingly urgent.

Conventional palletizing systems rely heavily on fixed-position robots programmed for specific tasks in highly structured environments. While such systems are robust and repeatable, they lack the flexibility to handle product variations, uncertain object poses, or dynamic arrangements. In many small and medium-sized enter–prises (SMEs), these limitations often result in the un–derutilization of automation, particularly when high product diversity, frequent changeovers, or space const–raints are present. Consequently, a new class of palleti–zing solutions is emerging, centered around lightweight robotic arms equipped with intelligent perception and affordable control platforms, designed to meet the chal–lenges of real-world industrial deployment.

Robotic arms, especially those with four to six deg–rees of freedom (DOF), offer a compact, flexible, and configurable solution for automated palletizing. Their compact design allows for easy installation in limited spaces, while their dexterity provides for handling a wide variety of packaging shapes. However, developing a flexible palletizing system requires a combination of real-time perception, dynamic decision making, and intelligent control. These capabilities are typically associated with high-end systems that require expensive sensors and computing platforms. Closing this gap requires a new approach that combines robust industrial controls with advanced computer vision and machine learning-based intelligence, all within a scalable and cost-effective framework.

Many recent studies have focused on the design and control of robotic arms for low-cost pick-and-place operations or automated industrial applications. Al

Received: April 2026, Accepted: May 2026
Correspondence to: Dang Anh Duy, Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Vietnam
E-mail: aduy@hcmut.edu.vn
doi: 10.5937/fme2603421P

Tahtawi et al. [1] developed an Arduino-controlled 3-degree-of-freedom robotic arm that uses inverse kine–matics to perform object picking and dropping accor–ding to input coordinates. Although basic experimental results were achieved, the positioning error remained high (~5 cm), which was insufficient to meet industrial requirements. Similarly, Cong [2] proposed a PLC-controlled SCARA robot that enables synchronization of stepper motor motion through high-speed pulse generation; however, it still operates based on preset logic and cannot self-recognize objects. At a higher mechanical level, Han Sol Kim et al. [3] designed a 5-degree-of-freedom robot for feeding/feeding workpieces on a lathe, combining kinematic modeling and structural analysis to ensure rigidity and accuracy. Meanwhile, Prabhakar et al. [4] developed a 6-DOF robot that is remotely controlled via a mobile app, aiming for low-cost automation in small businesses. Despite numerous improvements in mechanical design and control, these systems still cannot sense their environment and adjust flexibly due to a lack of integration with computer vision or artificial intelligence. Regarding robot calibra–tion, Xu et al. [5] employed a meta-heuristic algorithm (modified Chicken Swarm Optimization) in conjunction with repeatability testing to identify geometric parameters, while Cheng et al. [6] proposed a rotary axis fitting technique for calibrating robots according to ISO 9283:1998. Although these methods bring signi–ficant improvements in positioning accuracy, they still require manual calibration procedures and do not res–pond to changes in the actual operating environment. Other studies focused on optimizing the operation and placement of robots in the workspace. Wachter et al. [7] proposed an algorithm for optimizing the robot base position to reduce cycle time and energy consumption in a manufacturing environment. However, it still assumed a fixed input and could not sense the environment. In addition, Han et al. [8] studied the design of a desktop robot that eliminated the traditional reducer with a syn–chronous belt mechanism, which helped optimize cost and size, but only achieved precise control in a static environment. Overall, the above studies have made significant contributions to the development of simple, low-cost industrial robot systems; however, common limitations still exist, including the lack of integration between computer vision and artificial intelligence to recognize, locate, and adapt in complex environments.

The integration of computer vision and artificial intelligence technologies into industrial robotic systems enhances the ability to perceive, adapt, and make de–cisions in unstructured work environments. Instead of relying on fixed scripts and purely kinematic models, modern robotic systems are equipped with vision mo–dules that enable real-time object recognition, location, and gripping pose calculation. However, deploying computer vision in real industrial scenarios presents numerous challenges, including varying lighting condi–tions, partial occlusions, reflective surfaces, and the need for real-time performance, which are common obstacles that degrade the reliability of traditional vision systems.

To address these challenges, researchers and en–gineers have explored various vision-based object detection methods. Early approaches employed hand–crafted features, geometric models, or machine learning models, which suffered from poor generalization [9-11]. Nguyen et al. [12] developed a SCARA robot system that incorporates a least squares algorithm to estimate 3D coordinates from RGB images, thereby simplifying camera calibration and achieving an average error of less than 1 mm. However, the method still relies on color detection and cannot handle occluded or complex-shaped objects. Elassal et al. [13] designed a low-cost delta robot using Raspberry Pi to recognize objects through color and contour segmentation. The system is simple and easy to deploy, but it has low reliability in variable lighting environments and with objects of diverse colors. Similarly, Phuong et al. [14] also developed a delta robot that integrates vision to estimate object size and adjust the appropriate gripping distance. Although the results are accurate, the system cannot still learn and does not adapt well to unseen objects. Song et al. [15] proposed a method to correct the deburring trajectory using a CAD model combined with ICP and impedance control. The method ensures good accuracy and force control, but depends on the availability of a CAD model and a manual point teaching process. Zheng [16] and Singh [17] proposed image processing algorithms based on weighted contour matching and layered plane matching, respectively, to improve localization in monocular vision systems. Both of them help increase localization accuracy without the need for deep learning models, but are still susceptible to geometric noise and are ineffective in complex visual environments. Yang et al. [18] developed a tea picking robot system using YOLOv3 to detect tea buds and SVM optimized by PSO to determine picking points; however, the model requires manual calibration and has not been tested in complex field conditions. Similarly, Massah et al. [19] applied SVM, combining image features such as HOG, LBP, and histogram, to estimate kiwi yield, achieving a higher R^2 than AlexNet and ResNet; however, it still depends heavily on image quality and is difficult to adapt to fluctuating natural light environments. Zhang et al. [20] proposed a method for pomegranate recognition and counting based on 3D point cloud data and shape features, utilizing SVM and clustering algorithms. Although it achieves high accuracy in natural agricultural environments, this approach still relies on the integrity of the point cloud data and is inefficient when there are many occluded objects.

Later, convolutional neural networks (CNNs) revo–lutionized object detection with models such as YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), and Faster R-CNN [21-25]. Jhang et al. [26] proposed a YOLOv4-based model, combined with Uni–form Experimental Design (UED), for robotic gripping under an IoT framework. By offloading computation to a central server, their system reduced hardware costs while improving mean average precision (mAP) to 95%, outperforming the baseline YOLOv4 by 7–10%. Similarly, Panigrahi [27] implemented a YOLOv4-tiny-based pick-and-place robot using the NVIDIA Jetson Nano platform. The system achieved 40 FPS real-time inference with improved positional accuracy, demon–strating the potential of lightweight models in low-cost deployments. Santos et al. [28] extended the use of YOLOv5 in a multifunctional robotic architecture that integrated object detection, product quality inspection via CNN-SVM, and a reconfigurable gripper—all managed through a digital twin framework. Their design emphasized adaptability without hardware changes. Meanwhile, Silveira [29] leveraged YOLOv8 for 3D vision in logistics, combining object and keypoint detection to support complex manipulation tasks. The system achieved a mAP of 98.3% with 96.4% precision and 95.6% recall, though it required extensive annotated data and computing resources. These models significantly improved accuracy and speed, making them suitable for many robotic applications. These studies confirm the effectiveness of YOLO models in robotic automation, particularly for real-time perception on resource-constrained platforms. Zhen Li [30] further expanded the application of YOLO by proposing a multirobot SLAM-based navigation and grasping framework, where YOLOv4 is combined with a generative grasping CNN (GGCNN) to accurately detect and grasp unknown objects in unstructured environments. Their mobile robotic arm system achieved a grasp success rate of 86%, highlighting the viability of two-stage perception and action pipelines in real-world navigation tasks. Additionally, Cheng-Jian Lin [31] developed a YOLO-based robotic inspection system capable of detecting dynamic objects on a conveyor, estimating object sizes using an ANN, and identifying defects via a convolutional fuzzy neural network (CFNN). Their CFNN achieved an F1-score of 0.9535 and outperformed models such as AlexNet and LeNet in both accuracy and model efficiency. These studies confirm the versatility and effectiveness of YOLO-based models across a range of robotic tasks, including industrial pick-and-place, quality inspection, and mobile manipulation. However, common limitations remain, including challenges in detecting small or occluded objects, dependence on high-quality datasets, and the need for extensive calibration and system integration when combining multiple neural architectures.

Recently, a new generation of object detectors based on transformer architectures has emerged, led by the pioneering DETR (Detection Transformer) model. DETR replaces the CNN backbone and anchor-based detection head with an end-to-end attention mechanism, allowing it to learn global relationships and directly predict object bounding boxes and labels [32-25]. While DETR shows promising results in general object detection, it struggles with small objects and slow convergence. To overcome these limitations, the Region-based Fully Deformable Transformer (RF-DETR) was proposed, enhancing DETR's architecture by incorporating deformable attention modules and region refinement techniques. RF-DETR enhances detection accuracy, particularly for small and occluded objects, while preserving the transformer's key advantages: end-to-end training, anchor-free prediction, and the elimination of NMS [36].

In this work, we develop a cost-effective robotic palletizing system that combines compact mechanical design, industrial-grade motion control, and transformer-based deep learning for object detection. The robot arm features a 4-DOF configuration, driven by AC servo motors and harmonic gearboxes, which delivers high positional accuracy with low mechanical complexity. The system is controlled by a Mitsubishi FX5U PLC, which offers reliable and deterministic execution while minimizing costs compared to traditional robotic controllers.

To enhance flexibility and grasp precision, we integrate a deep learning-based vision module built on RF-DETR. This transformer-based network is trained on a custom dataset of retail and logistics items, predicting object bounding boxes and orientations in 2D space. Using depth information from an Intel RealSense RGB-D camera, these 2D predictions are projected into 3D grasp coordinates via camera intrinsics and per-pixel depth values. A calibrated rigid transformation converts these into robot coordinates for precise actuation.

By combining deep learning-based object detection with real-time 3D sensing, our system can robustly detect and grasp objects under challenging conditions, such as occlusion, stacking, and varied packaging shapes. The proposed architecture demonstrates that advanced deep learning perception can be effectively deployed in low-cost robotic systems, enabling adaptive and accurate manipulation for industrial applications.

The contributions of this work can be summarized as follows:

- Present a complete design and implementation of a 4-DOF palletizing robot system, integrating mechanical structure, PLC-based motion control, and real-time vision processing using RF-DETR and depth sensing.
- Adapt the RF-DETR architecture to detect and localize a diverse set of retail and logistics items with varying shapes, colors, and materials, and convert image-based grasp points into 3D robot coordinates through calibrated geometric transformation.
- Integrate AI-based perception with PLC-level control in real time, and conduct comparative evaluation against YOLOv12 and other state-of-the-art models, demonstrating the superior robustness of RF-DETR, particularly under partial occlusion.

The remainder of this paper is organized as follows: Section 2 presents the mechanical and electrical design of the robot system. Section 3 describes the kinematic modeling and inverse kinematics of the robot arm. Section 4 introduces the vision system based on RF-DETR and RealSense-based 3D perception. Section 5 discusses the experimental setup, evaluation metrics, and results. Finally, Section 6 concludes the paper and outlines directions for future research

2. DESIGN AND PERFORMANCE OF 4-DOF PALLETIZING ROBOT ARM

2.1 Mechanical design

The mechanical design of the palletizing robot arm focuses on achieving high precision and stability to meet the requirements of industrial applications. Figure 1 illustrates the robot's design in Autodesk Inventor software. The robot has 4 degrees of freedom (DOF), using AC servo motors attached to harmonic gearboxes at each joint, as shown in Figure 2. Tables 1 and 2 list the specifications for the harmonic gearbox and servo motors, respectively.

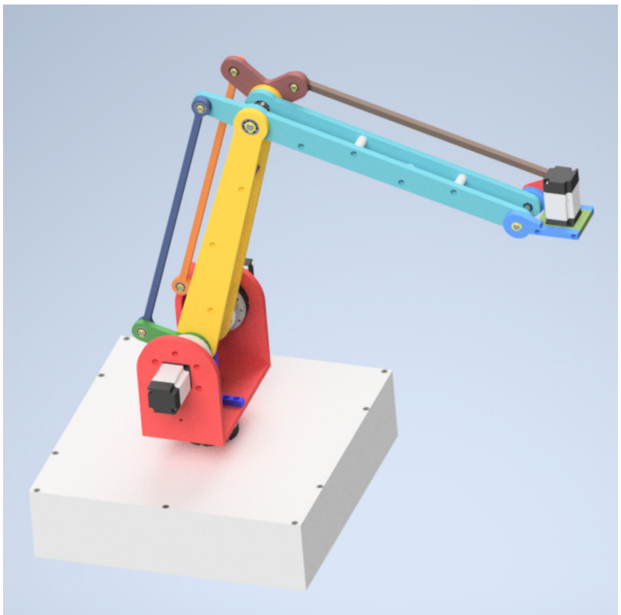


Figure 1. The design of the 4-DOF Palletizing robot arm on the Autodesk Inventor software

Each joint of the robot arm is driven by an AC servo motor (Yaskawa SGM7J-01AFC6S) with a rated power of 100 W, a rated torque of 0.318 Nm, and a maximum torque of 1.11 Nm. The motor can rotate at a maximum speed of 3000 RPM. A 24-bit encoder mounted at the end of the motor provides precise position feedback to the motor. To increase torque and reduce backlash, each motor is coupled to a harmonic gearbox (Model: CSF-17-50-2UH) with a gear ratio of 1:50. The harmonic gearbox can operate for a long time with a rated torque of 16 Nm, an average torque limit of 26 Nm, and a maximum torque of 34 Nm. This helps to ensure stability under various load conditions. The motor-gearbox combination provides the torque and precision needed for the robot's joints.

Table 1. The specifications of harmonic gearboxes

Specification	Value
Model	CSF-17-50-2UH
Gear ratio	1:50
Rated Torque	16 Nm
Average Torque Limit	26 Nm
Maximum Torque Limit	34 Nm
Starting Torque	6.1 Nm
Input Speed	3500rpm
Basic Dynamic Load	52.9×10^2N
Basic Static Load	75.5×10^2N
Weight	0.68kg

Table 2. The specifications of Yaskawa servo motors

Specification	Value
Rated Power	100W
Rated Torque	0.318 Nm
Maximum Torque	1.11 Nm
Rated Speed	3000 rpm
Encoder Resolution	24-bit
Rated Current	0.85 A

Figure 3 illustrates the structure of the robot base. The base of the robot arm is a stable platform that supports the entire upper body of the robot. The AC motors and harmonic gearboxes are fixed in the base and provide motion for the robot's links. The motor for the first joint is fixed in a vertical position, allowing the upper body of the robot to rotate around a vertical axis. The other two motors are arranged horizontally on both sides, providing motion for the second and third links of the robot. The second link is directly connected to the gearbox of the second motor, while the third link receives motion through the connecting rod in the four-bar parallelogram mechanism. The robot end effector is held parallel to the horizontal plane by two paral–lelogram mechanisms, as shown in Figure 4. This is essential for palletizing applications, where the end ef–fector is held horizontally to stack packages.

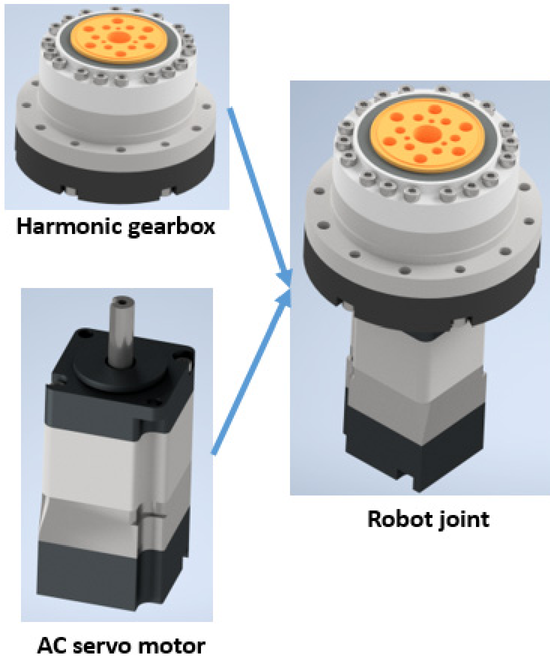


Figure 2. The structure of a robot joint

The mechanical parts of the robot arms are manu–factured from 6061 aluminum alloy to minimize the mass of the arms while maintaining rigidity. Reducing the mass of the arms reduces the moment of gravity at each joint, thereby increasing the robot's payload. The mechanical parts are manufactured in the form of 10 mm thick plates and then assembled into the robot's links.Using metal plates saves costs and processing time compared to processing links from a large block of metal.Each link of the robot arm measures 400 mm in length, providing a total reach of 800 mm. The robot is designed to carry a payload of up to 1 kg. The total weight of the structure is approximately 6 kg, providing an optimal balance between rigidity and lightness.

The electrical system of the palletizing robot is designed to control the robot arm's movement accurately and efficiently. The system uses the Mitsubishi FX5U-32MT/ES PLC to control the operation of all components in the circuit. Figure 5 illustrates components of electrical devices in the electrical system. Table 3 lists the key elements of the electrical system.

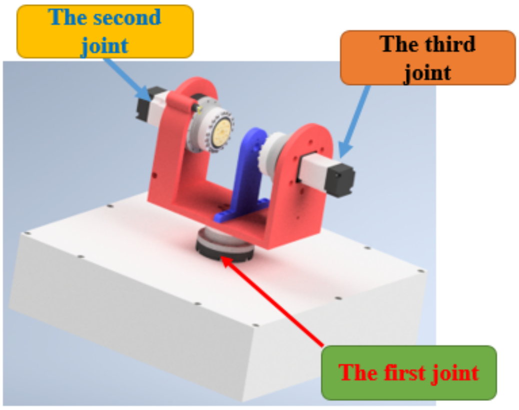


Figure 3. The structure of the robot base

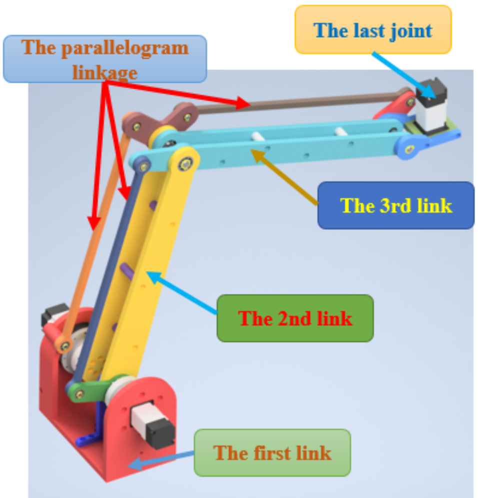


Figure 4. The structure of the robot base

2.2 Electrical system design using PLC

The Mitsubishi FX5U-32MT/ES PLC is employed as the core controller of the electrical system. It has 16 input pins and 14 output pins, with four high-speed pulse output pins. The PLC processes input signals from proximity sensors and ON/OFF push buttons, and outputs high-speed pulse signals to the motor drivers to control the motion of the four servo motors.

The SGD7S-R90A00A002 motor drivers receive pulse signals from the Mitsubishi FX5U-32MT/ES PLC and convert them into the necessary motor commands. These drivers ensure smooth and precise rotation moti–ons of the servo motors, which drive the robot's joints.

Hanyoung UP18RD-8NA proximity sensors are used to limit the rotation angles of the robot's joints and determine the joints' initial position. These sensors are connected to the PLC's inputs. When the robot is started, the PLC will control the motors for moving the robot's joints to touch the proximity sensors (and then halt). The controller determines this position as the initial angle for the joints. Moreover, during operation, when the joints come into contact with the sensors, the motors will also halt to ensure that the robot operates within its mechanical limits.

To communicate between the PLC and the com–puter, a USB-to-RS-485 converter is used. A program is written on the computer to control the robot's operation. The program interface will include buttons to move the robot's joints. The software also sets parameters such as maximum speed and rotation angle resolution. When there is an impact or change in parameters on the computer screen, the program generates corresponding codes and sends them to the PLC via RS-485 communication. Additionally, the computer program reads and processes the images captured by the 3D camera to extract the coordinates of the objects of interest, converts them to the number of pulses for the motors, and sends the information to the PLC. The computer program will control the entire movement process of the robot through RS-485 communication. When the robot completes a movement, it sends back a corresponding code, allowing the computer to recognize it and send the next execution command.

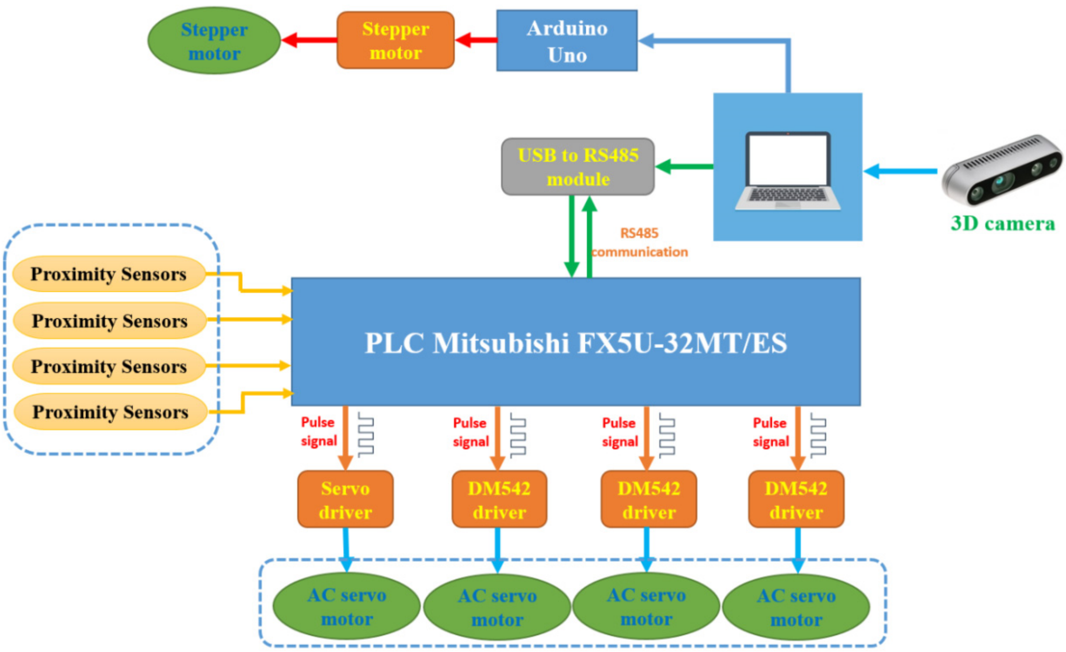


Figure 5. The diagram of the electrical system

Table 3. Key Components of the Electrical System

Component	Specification
PLC	Mitsubishi FX5U-32MT/ES, 16 Inputs, 16 Outputs, four high-speed pulse outputs
Servo Motor Driver	SGD7S-R90A00A002, 200-240 VAC, 50Hz/60Hz, 0.1 kW
Servo Motors	Yaskawa SGM7J-01AFC6S, 100W, 3000 RPM
Proximity Sensors	Hanyoung UP18RD-8NA, NPN, 8 mm sensing distance
Arduino Uno	ATmega328P microcontroller, 14 digital I/O pins (6 PWM), six analog inputs, 16 MHz clock speed
Stepper motor driver	DM542, 1.0–4.2A adjustable current, microstepping up to 1/128, 18–50V DC input voltage range
Stepper motor	NEMA 17, 1.8° step angle, 1.5A phase current, holding torque ~45 N·cm
Status Indicator Lights	ON/OFF, controlled by PLC.
Push Buttons	ON/OFF, manual control
Power Filter	**ZAC 2210-11**, filters noise from the AC supply
Intermediate Relays	**24V**, used for handling high-current loads
Power Supply	24V DC (PLC and sensors), 200-240V AC (motor drivers and motors)
Safety Mechanisms	E-Stop, overload protection, and proximity sensors for joint limits

Because the Mitsubishi FX5U-32MT/ES PLC has only 4 high-speed pulse outputs, the gripper utilizes a stepper motor to provide movement. Therefore, an Arduino Uno board is used to control the stepper motor. When the robot has moved to the position to grip the object, the computer sends a signal to the Arduino to control the stepper motor, causing the gripper to close.

In addition to the main components mentioned above, the circuit also has additional components used to supply power and protect the circuit. The electrical system includes a 200-240 VAC power supply for the motor drivers and a 24V DC power supply for the Mitsubishi FX5U-32MT/ES PLC, sensors, and relays. The ZAC 2210-11 power filter is installed to ensure a clean power supply, filtering out electrical noise and protecting sensitive components. Safety mechanisms include:

Emergency Stop (E-Stop): A manual failsafe that immediately cuts power to the system in the event of an emergency.

Overload Protection: Circuit breakers and fuses are installed to prevent damage from power surges or overcurrent conditions.

Power Filter (ZAC 2210-11): The ZAC 2210-11 power filter is installed to ensure the system receives clean, stable power by filtering out noise or voltage spikes from the AC power supply. This helps protect sensitive electronic components, such as motor drivers and PLCs, from damage caused by power disturbances.

ON/OFF Push Buttons: The ON/OFF push buttons allow manual control of the system. The PLC processes inputs from these buttons to turn the system on or off in response to the operator's actions.

Status Indicator Lights (ON/OFF): These indicator lights, controlled by the Mitsubishi FX5U-32MT/ES PLC, provide visual feedback on the robot's operational status. They signal whether the system is powered on (ON) or off (OFF).

3. DESIGN AND PERFORMANCE OF 4-DOF PALLETIZING ROBOT ARM

Figure 6 shows the kinematic diagram of the robot arm. The robot consists of four rotation joints. The first and last joints rotate around the vertical axis, while the 2nd and 3rd joints rotate around the horizontal axis. Assuming the base frame is at the first point of the robot, the Z-axis is oriented upward. The end-effector coordinate frame is at the end of the last link. Because the robot's links are connected through parallelogram mechanisms, the robot's end-effector is always parallel to the robot's base frame.

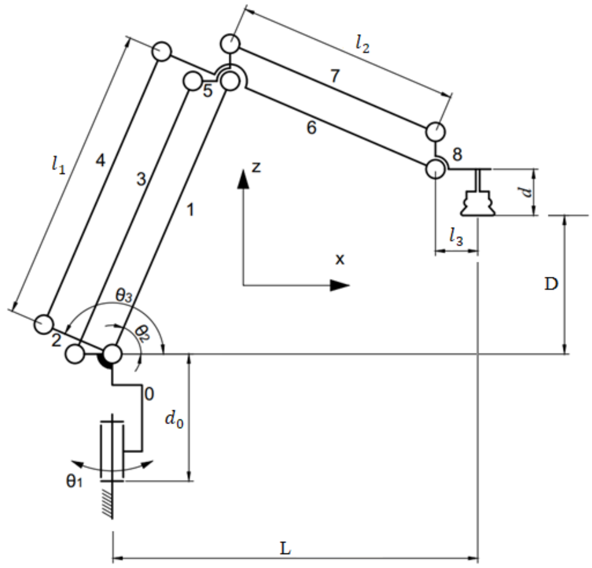


Figure 6. The kinematic diagram of the robot arm

This section analyzes the geometric relationships of the robot base frame and the end-effector frame. In Figure 6, the distance from the robot base to the 2[nd] joint's axis is d_0. The lengths of the links are l_1, l_2 and l_3, respectively. The distance from the last joint's axis to the robot end-effector is l_4. Based on the geometric relationship, the distance between the first joint and the last joint in the horizontal plane is:

$$L = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \tag{1}$$

and the distance according to the Z-axis direction is:

$$D = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) - d \tag{2}$$

In the XY plane, the coordinate of the robot end-effector is:

$$\begin{aligned} X &= L\cos\theta_0 + l_4 \cos(\theta_0 + \theta_3) \\ Y &= L\sin\theta_0 + l_4 \sin(\theta_0 + \theta_3) \end{aligned} \tag{3}$$

According to the Z-axis, the Z-coordinate of the end-effector is:

$$Z = d_0 + D \tag{4}$$

Submitting the values of L and D to Equations (3) and (4), the coordinates of the robot end-effector are:

$$X = \left(l_1 \cos\theta_1 + l_2 \cos\left(\theta_1 + \theta_2\right) + l_3\right)\cos\theta_0 + l_4 \cos\left(\theta_0 + \theta_3\right) \tag{5a}$$

$$Y = \left(l_1 \cos\theta_1 + l_2 \cos\left(\theta_1 + \theta_2\right) + l_3\right)\sin\theta_0 + l_4 \sin\left(\theta_0 + \theta_3\right) \tag{5b}$$

$$Z = d_0 + l_1 \sin\theta_1 + l_2 \sin\left(\theta_1 + \theta_2\right) - d \tag{5c}$$

The end-effector only rotates around the Z-axis, so the orientation angle of the robot end-effector is:

$$\theta_e = \theta_0 + \theta_3 \tag{6}$$

Equations (5) and (6) are used in the case that we want to find the position and orientation of the end-effector if we have the rotation angle of each joint. However, in real applications, the inverse problem is often used to determine the rotation angles of the joints and control the robot to the desired position.The solution to the inverse kinematics problem can be found by solving the forward equation.

From Equations (5a) and (5b), the value of the first joint angle can be easily obtained:

$$\theta_0 = atan\left(\frac{Y - l_4 \sin\theta_p}{X - l_4 \cos\theta_p}\right) \tag{7}$$

Then, the value of the last joint angle is also calculated from Equation (6):

$$\theta_3 = \theta_e - \theta_0 \tag{8}$$

From Equations (5) and (6), we eliminate the angles θ_0 and θ_3 by converting and summing the squares of the first two equations. Denote:

$$a = \pm\sqrt{\left(X - l_4 \cos\theta_e\right)^2 + \left(Y - l_4 \sin\theta_e\right)^2} \tag{9}$$

$$b = Z - d_0 + d \tag{10}$$

Equation (5) is transformed into:

$$l_1 \cos\theta_1 + l_2 \cos\left(\theta_1 + \theta_2\right) = a - l_3 \tag{11a}$$

$$l_1 \sin\theta_1 + l_2 \sin\left(\theta_1 + \theta_2\right) = b \tag{11b}$$

Summing the square of Equation (11) obtain:

$$\cos\theta_2 = \frac{\left(a - l_3\right)^2 + b^2 - l_1^2 - l_2^2}{2 l_1 l_2} \tag{12}$$

The value of the angle θ_2 is:

$$\theta_2 = \pm\arccos\left(\frac{\left(a - l_3\right)^2 + b^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \tag{13}$$

After the value of the angle θ_2 is determined, Equation (11) only contains the variable θ_1. Expand the Equation (11a) according to the term θ_1:

$$\left(l_1 + l_2 \cos\theta_2\right)\cos\theta_1 - l_2 \sin\theta_2 \sin\theta_1 = a - l_3 \tag{14}$$

Denote:

$$\begin{aligned} r &= \sqrt{\left(l_1 + l_2 \cos\theta_2\right)^2 + \left(l_2 \cos\theta_2\right)^2} \\ \varphi &= atan2\left(l_2 \sin\theta_2, l_1 + l_2 \cos\theta_2\right) \end{aligned} \tag{15}$$

Equation (14) becomes:

$$\cos\left(\theta_1 + \varphi\right) = \frac{a - l_3}{r} \tag{16}$$

Finally, the angle θ_1 is:

$$\theta_1 = \pm\arccos\frac{a - l_3}{r} - \varphi \tag{17}$$

4. OBJECT DETECTION AND 3D GRASP LOCALIZATION USING RF-DETR IN ROBOTIC PALLETIZING

4.1 Object detection using RF-DETR.

To achieve robust and accurate object detection in cluttered and partially occluded palletizing environments, this study employs RF-DETR (Roboflow Detection Transformer) as the object detection backbone. RF-DETR is a real-time detection transformer developed by Roboflow and built on a DINOv2 vision transformer backbone. As a member of the DETR family, RF-DETR performs end-to-end object detection without requiring hand-designed post-processing such as non-maximum suppression, thereby simplifying the inference pipeline while maintaining strong detection accuracy.

RF-DETR has been reported to provide a favorable trade-off between detection accuracy and inference latency on benchmark datasets such as MS COCO and RF100-VL, making it suitable for practical industrial vision applications where both accuracy and responsiveness are required. Another advantage of RF-DETR is its strong adaptability to custom datasets, which is particularly important in robotic palletizing scenarios involving domain-specific object categories, partial occlusion, pose variation, and non-uniform illumination.

In our system, RF-DETR is trained on a custom dataset comprising eight distinct object categories commonly encountered in palletizing tasks. For each object category, approximately 200 RGB images were captured under different poses, lighting conditions, and background settings. Figure 7 presents representative images from the custom dataset. These raw images were uploaded to Roboflow for annotation, where each object instance was labeled with a bounding box corresponding to its class. To improve robustness and generalization, several augmentation techniques, including flipping, rotation, brightness variation, and cropping, were applied. As a result, the effective dataset size increased to approximately 600 annotated samples per class, yielding a total of about 4,800 training images.

Annotation was carried out manually using Roboflow's web-based interface to ensure accurate bounding boxes and class labels. The annotated dataset was then exported in the required format and used to fine-tune RF-DETR in Google Colab. A pretrained checkpoint was used to initialize the network and accelerate convergence during fine-tuning. Hyperparameters such as learning rate, batch size, and number of epochs were selected experimentally to achieve a balance between convergence speed and model generalization. During training, the optimization follows the standard DETR-style formulation, combining classification and boun–ding-box regression objectives with bipartite matching between predictions and ground-truth annotations. Mo–del performance was continuously monitored using validation mAP and loss curves to reduce overfitting and ensure stable training.

After training, the optimized RF-DETR model was deployed for real-time inference using RGB images acquired from the Intel RealSense camera. The detector outputs object class labels and 2D bounding box coordi–nates, which are subsequently used to estimate grasp-related geometric information. These 2D observations are then integrated with depth information to determine the corresponding 3D grasp location, as described in Section 4.2. In this way, RF-DETR serves as the key perception module that enables reliable object localiza–tion under overlapping layouts, changing illumination, and partial occlusion, which are common challenges in robotic palletizing systems.

Figure 7. The images in the custom dataset

To enhance training stability and improve generali–zation, we employ an Exponential Moving Average (EMA) model in conjunction with the base RF-DETR architecture. The EMA model maintains a shadow copy of the base model's weights, which are updated iteratively using an exponential moving average scheme during training. Specifically, the EMA weights $\theta_{EMA}^{(t)}$ at epoch t are updated according to:

$$\theta_{EMA}^{(t)} = \alpha \cdot \theta_{EMA}^{(t-1)} + (1-\alpha) \cdot \theta(t) \tag{18}$$

where $\theta^{(t)}$ denotes the base model weights, and α is a smoothing factor typically set close to 1 (e.g., 0.999). Unlike the base model, the EMA model is not used for gradient updates; instead, it is evaluated periodically on the validation set. This approach mitigates the effects of noisy weight updates and model oscillation during trai–ning, leading to more stable and often more accurate predictions. As demonstrated in our experiments, the EMA-enhanced RF-DETR consistently outperforms the base model in both average precision and recall across various IoU thresholds.

4.2 2D-to-3D Grasp Point Transformation

To enable accurate robotic manipulation, it is essential to convert the 2D grasp predictions obtained from the RF-DETR model into actionable 3D coordinates in the robot's working space. This transformation process involves both intrinsic and extrinsic calibration para–meters and requires synchronization between the RGB and depth streams provided by the Intel RealSense ca–mera. Figure 8 illustrates the flowchart of this process.

The grasp parameters predicted by the RF-DETR model include the center position (x,y) of the bounding box and its dimensions (width, height). These parameters are initially expressed in pixel coordinates relative to the image frame. The Intel RealSense RGB-D camera provides per-pixel depth values, which are used to determine the spatial location of the grasp point in the 3D space.

The first step involves retrieving the depth value z at the pixel location (x,y) from the aligned depth image. With the known camera intrinsics (focal lengths f_x, f_y, and the principal point c_x, c_y), the 3D point (X_C, Y_C, Z_C) in the camera coordinate system is calculated using the pinhole camera model:

$$\begin{aligned} X_C &= \frac{x - c_x}{f_x} Z \\ Y_C &= \frac{y - c_y}{f_y} Z \end{aligned} \tag{19}$$

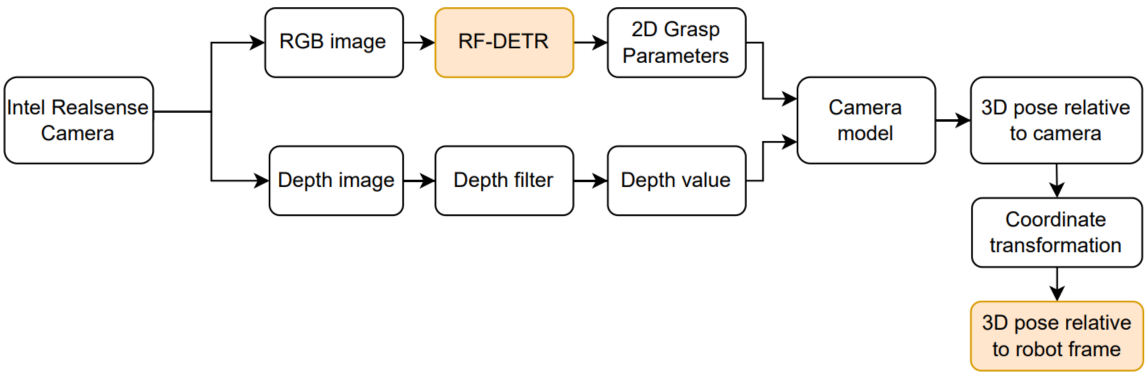


Figure 8. Flowchart of the 2D-to-3D Grasp Point Transformation

This step provides the 3D grasp point (X_C, Y_C, Z_C) in the camera's coordinate system. However, to execute the grasp, the 3D coordinates must be transformed into the robot's coordinate system. The transformation from the camera coordinate system to the robot's coordinate system requires the use of the camera's extrinsic parameters, which include the rotation matrix R and translation vector T. These extrinsic parameters describe the camera's position and orientation relative to the robot base frame. The transformation matrix [R|T] allows the conversion of the 3D grasp point from the camera's frame to the robot's frame using:

$$P_r = T_r^c P_c \tag{20}$$

where $P_c = [X_C, Y_C, Z_C, 1]^T$ is the grasp point in the camera coordinate system, and $P_r = [X_r, Y_r, Z_r, 1]^T$ is the grasp point in the robot's coordinate system. This matrix multiplication ensures that the grasp point is accurately represented in the robot's working coordinate frame. Figure 9 shows the relationship of the coordinates.

Once the 3D coordinates have been transformed, the computer computes the necessary rotation angle for the robot's end-effector based on the orientation θ predicted by the CNN. This ensures that the gripper is aligned with the object correctly before performing the grasp. These 3D coordinates and rotation commands are sent to the Programmable Logic Controller (PLC), which drives the robot's motors to move the end-effector to the calculated position and orientation.

To reduce the impact of noisy or missing depth values, a local neighborhood median filter is applied when retrieving depth at (x, y). Furthermore, the transformation pipeline is validated using calibration targets and known ground truth positions, ensuring sub-centimeter accuracy in the final 3D grasp point estimation.

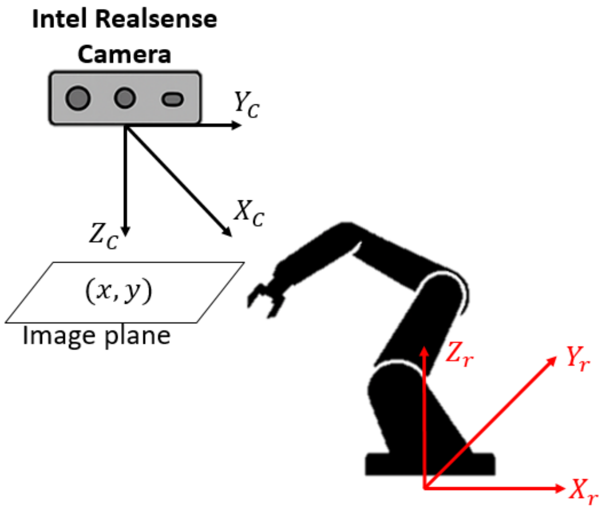


Figure 9. Coordinate transformation from the image plane to the robot frame.

In essence, the computer vision system seamlessly integrates image capture, neural network-based grasp detection, and coordinate transformation to provide the robot with real-time control over its movements. By utilizing intrinsic and extrinsic camera parameters, along with precise coordinate transformations, the system ensures that grasp points detected in 2D are accurately converted into 3D and translated into the robot's workspace, enabling efficient and reliable grasping operations in dynamic environments.

5. EXPERIMENT RESULTS AND DISCUSSIONS

Figure 10 illustrates the physical setup of the proposed intelligent palletizing system. The Intel RealSense D435 RGB-D camera is mounted approximately 1 meter above the conveyor belt to ensure a full top-down view of the robot's operating workspace. This positioning enables precise object localization while minimizing occlusions and avoiding interference with the robot's movement during execution. Depth information captured from this vantage point is critical for accurate 3D pose estimation of the grasping targets.

The object detection and grasp planning module is executed on a personal computer equipped with an NVIDIA Quadro M1000 GPU, which provides sufficient computational power to run the RF-DETR model in real-time. The 4-DOF robotic arm, custom-built for this study, is mounted on a rigid base and connected to a mechanical two-finger gripper, allowing for stable handling of objects with diverse shapes and sizes.

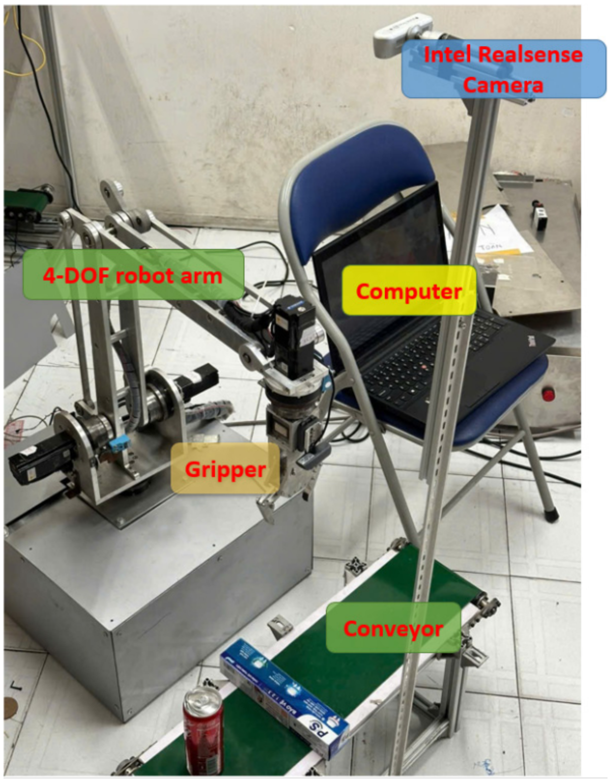


Figure 10. Experiment setup of the robot system.

A conveyor belt is used to simulate a dynamic production environment, enabling the system to evaluate detection and grasping performance under varying object positions and orientations. This setup provides a practical and reproducible testing environment that reflects the constraints of real-world factory operations. Furthermore, the modular configuration of the system allows for future integration with additional sensors, feedback control modules, or industrial PLCs, supporting its deployment in scalable automation scenarios.

5.1 RF-DETR Training and Evaluation

The training process was conducted using the AdamW optimizer with a learning rate of 1×10^{-4}, over a total of 15 epochs. A batch size of 16 was used with gradient accumulation (accumulation steps = 4). This confi–guration ensured stable convergence while maintaining computational efficiency on the available hardware.

Figure 11 illustrates the training progress of the RF-DETR model over 15 epochs, including training and validation loss, as well as average precision (AP) and average recall (AR) at different IoU thresholds. The training and validation loss curves exhibit a consistent downward trend, stabilizing after epoch 10, indicating good convergence without overfitting. In terms of de–tection performance, both the base model and the EMA-enhanced model achieved high AP and AR scores early in the training process. AP@0.5 exceeded 0.95 by epoch 4 and remained stable, while AP@0.5:0.95 reached approximately 0.78. Similarly, AR@0.5:0.95 plateaued around 0.88. The EMA model exhibited slightly better stability and precision across metrics, demonstrating its effectiveness in refining predictions during training.

The mean Average Precision (mAP) across different IoU thresholds and object sizes is illustrated in Figure 12. The model achieved an overall mAP@0.5:0.95 of 0.78, with high performance at mAP@0.5 (0.92) and mAP@0.75 (0.88), indicating reliable detection across varying localization strictness. When analyzed by object scale, RF-DETR showed the strongest performance on large objects (mAP@0.5:0.95 = 0.77, mAP@0.5 = 0.91, mAP@0.75 = 0.86), followed by medium-sized objects. In contrast, the model yielded low scores for small objects, likely due to the limited representation of these objects in the training dataset. These results confirm that RF-DETR is particularly well-suited for detecting me–dium to large objects commonly encountered in palle–tizing applications.

Figure 13 presents qualitative detection outputs. The RF-DETR model effectively detects both individual and multiple objects in diverse scenes, including cluttered environments, occlusions, and non-ideal lighting condi–tions. These results further confirm the model's robust–ness and its suitability for deployment in real-world palletizing scenarios.

Table 4. Performance comparison of RF-DETR and baseline models

Model	mAP @0.5:0.95	mAP @0.5	Inference Time (ms)
RF-DETR-B	0.78	0.92	6
LW-DETR-M	0.705	0.910	6
YOLOv12m	0.723	0.915	4
YOLOv11m	0.716	0.911	5.7
YOLOv8m	0.702	0.910	6.3

To further validate the effectiveness of the proposed RF-DETR architecture, we conducted a comparative study against several state-of-the-art object detection models, including YOLOv12, YOLOv11, YOLOv8, and LW-DETR. The results are shown in Table 4.

Although YOLOv12 achieves the fastest inference time (4 ms) among the evaluated models, RF-DETR-B consistently delivers the highest detection precision, particularly at higher IoU thresholds (mAP@0.5:0.95). Compared to other transformer-based detectors such as LW-DETR-M, RF-DETR-B improves the mAP@ 0.5:0.95 by over 7%, demonstrating its architectural advantage. Similarly, when compared to convolutional-based detectors like YOLOv11m and YOLOv8m, RF-DETR-B shows clear superiority in both overall accu–racy and robustness in complex scenes. These impro–vements can be attributed to RF-DETR's refined object-query mechanism and its attention-driven feature aggre–gation, which enable more effective reasoning under conditions of clutter, occlusion, and visual ambiguity.

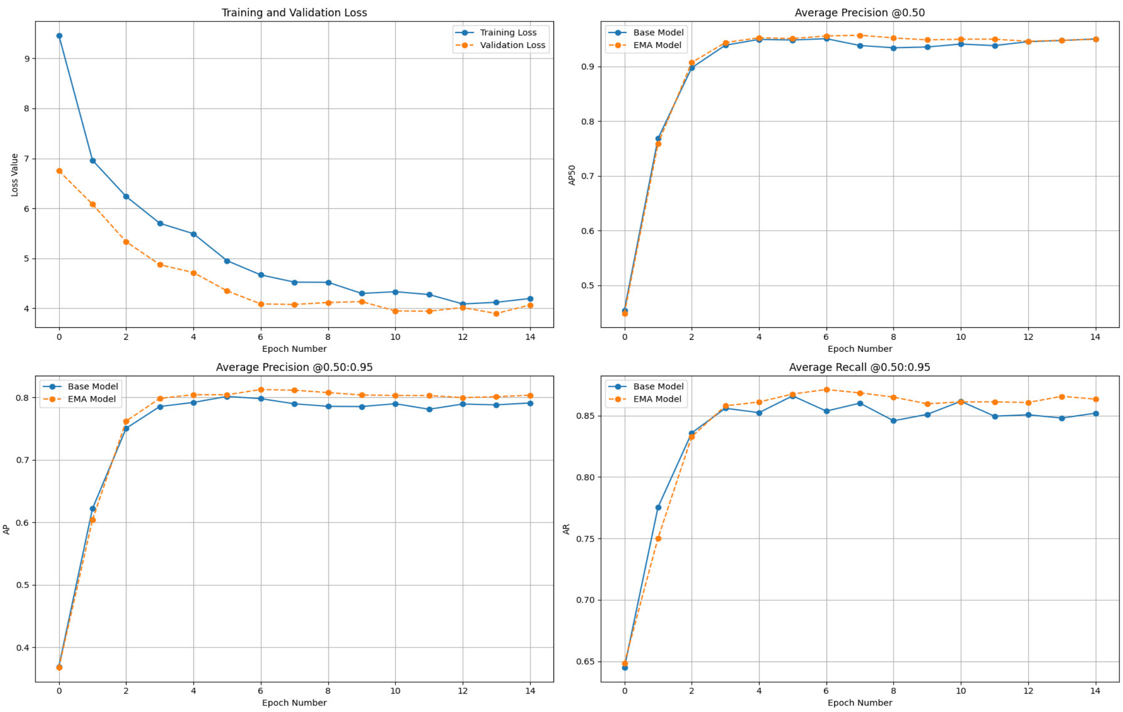


Figure 11. Training Progress and Evaluation Metrics of RF-DETR over 15 Epochs.

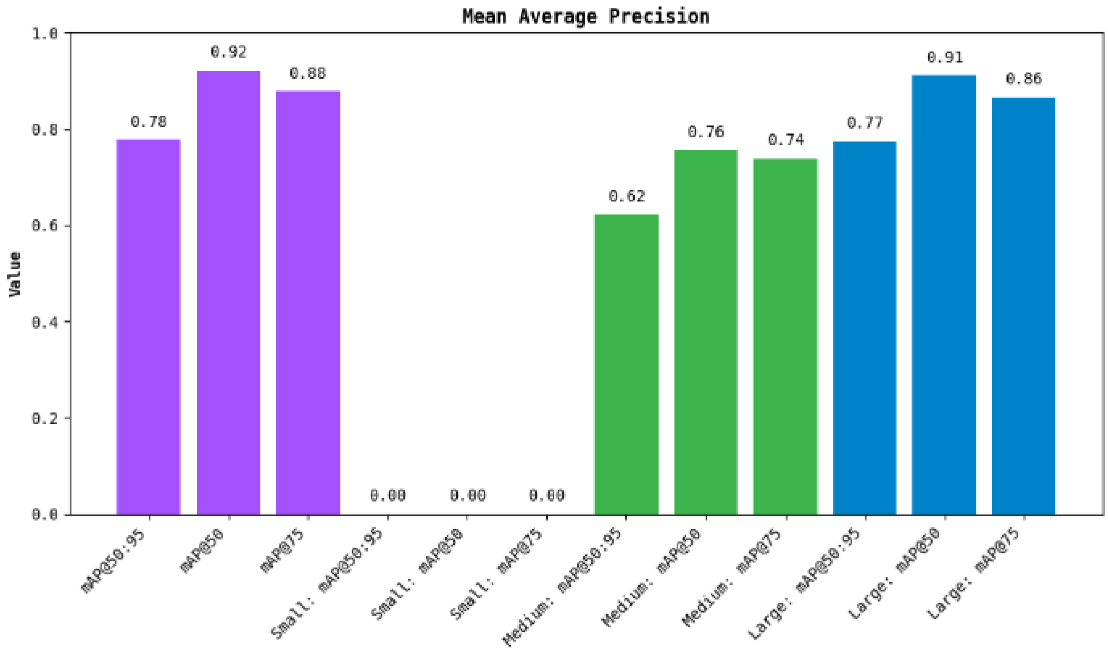
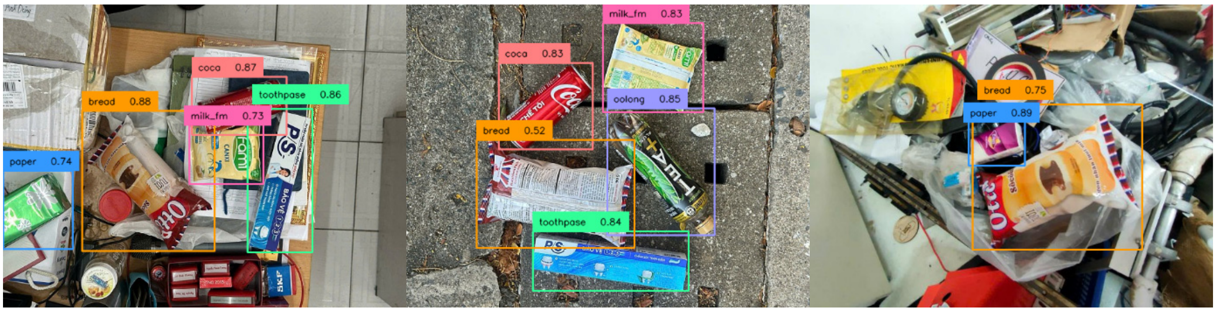
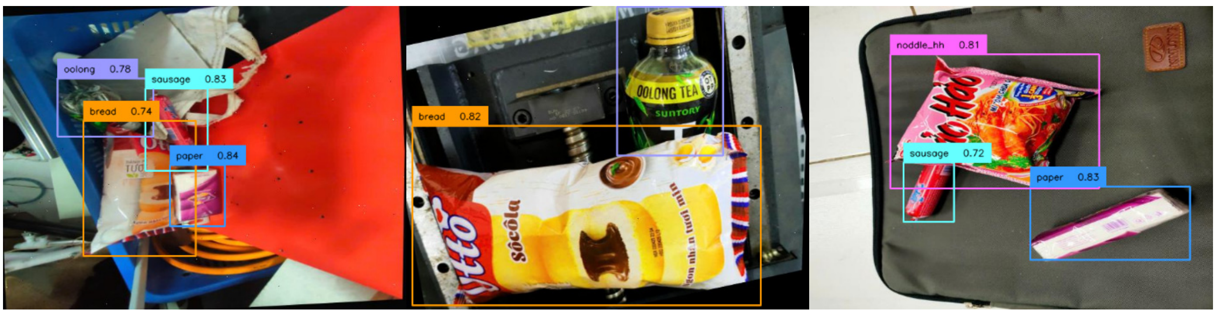


Figure 12. Mean Average Precision (mAP) of RF-DETR across object sizes and IoU thresholds



(a) normal lighting with multiple background textures

(b) challenging illumination conditions



(c) scenes with partial occlusion.

Figure 13. Detection results of RF-DETR under varying environmental conditions and diverse scenes.

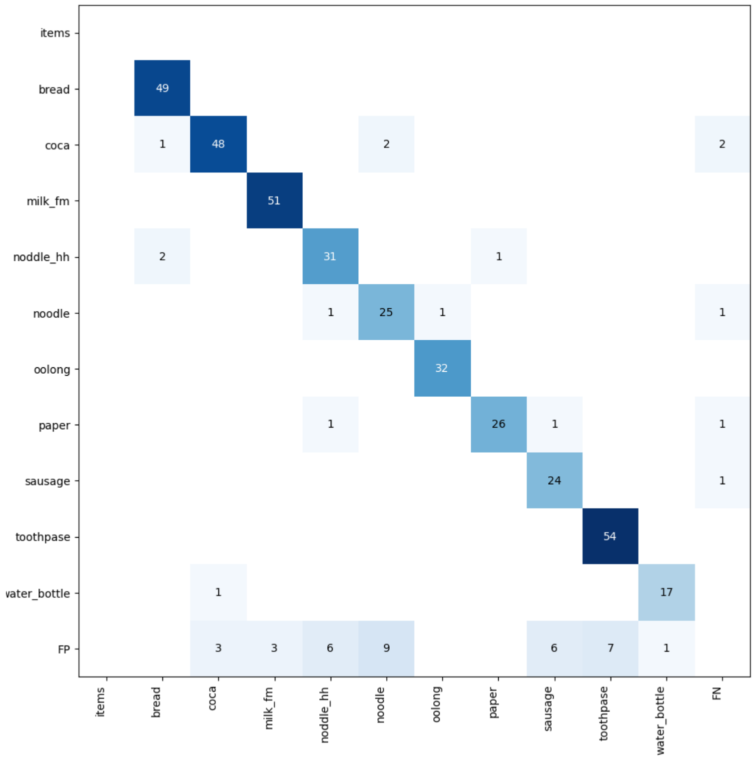


Figure 14. Confusion matrix of the RF-DETR model

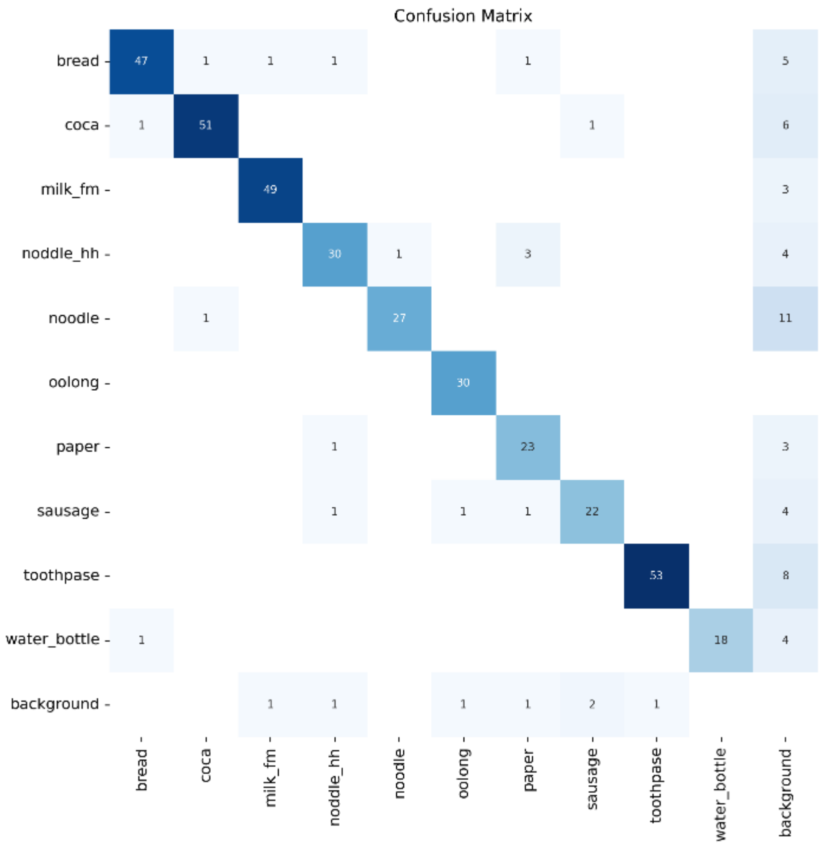


Figure 15. Confusion matrix of the YOLOv12 model

The confusion matrix in Figure 14 demonstrates that RF-DETR achieves high classification accuracy across most object categories, with strong diagonal dominance indicating consistent predictions. Classes such as *toothpaste*, *milk_fm*, and *bread* are recognized with near-perfect precision, while minor confusions are observed between visually similar items like *noodle* and *noodle_hh*. Despite a few false positives and negatives, particularly for *water_bottle*, the model maintains robust performance, confirming its suitability for object detection tasks in palletizing applications where fine-grained classification is required.

Figure 15 shows the confusion matrix of the YOLOv12 model. A comparison of the confusion matrices reveals that RF-DETR generally outperforms YOLOv12 in classification consistency and robustness. While both models exhibit strong accuracy on dominant classes such as *toothpaste* and *milk_fm*, RF-DETR achieves clearer diagonal dominance with fewer off-diagonal errors, particularly for visually similar categories such as *noodle* vs. *noodle_hh* and *paper* vs. *sausage*. YOLOv12, although slightly faster in inference, exhibits more confusion with background regions and a higher number of false positives in challenging cases, such as *water_bottle*.

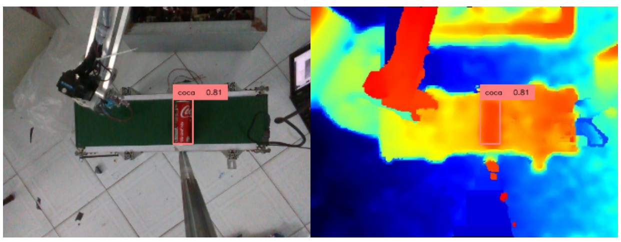
Beyond classification metrics, we also evaluated the models under occlusion conditions. As illustrated in Figure 16, YOLOv12 struggles to detect objects when more than 70% of their area is occluded, often failing to localize or classify them correctly. In comparison, RF-DETR demonstrates significantly better resilience, maintaining detection capability even under heavy occlusions. This superior robustness can be attributed to its attention-based transformer architecture, which enables global context reasoning and partial feature reconstruction. These findings further confirm that RF-DETR provides more reliable detection for palletizing applications where object visibility can be limited due to stacking or overlapping.

Figure 17 presents the qualitative results of the RF-DETR-based detection system integrated within a real-world conveyor-based palletizing environment. Each detection sample consists of a pair of images: the left image shows the RGB view captured from a top-mounted camera, while the right image displays the aligned depth map acquired from the same perspective. The system was evaluated on eight different object types commonly encountered in consumer packaging and logistics, including toothpaste, noodle packs, bread, water bottles, milk cartons, and other retail items. These objects were placed individually on the conveyor belt with varied orientations, including upright, tilted, and slightly rotated poses to simulate realistic placement scenarios.

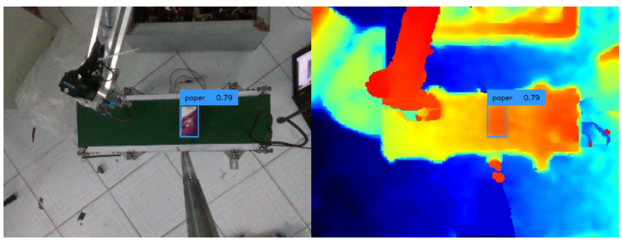
The RF-DETR model demonstrated robust detection performance across all object instances. Bounding boxes were consistently and accurately drawn on both RGB and depth images, with corresponding class labels and confidence scores clearly annotated. A minimum confidence threshold of 0.50 was set to qualify a detection as valid. In all cases presented, the confidence scores ranged from 0.53 to 0.90, indicating that the model not only met but significantly exceeded the ac–ceptance criterion. This confidence margin indicates the model's strong ability to differentiate object classes, even in the presence of slight noise or background variation.

a. RF-DETR model b. Yolov12 model

Figure 16. Detection comparison under occlusion conditions between YOLOv12 and RF-DETR.



Object 1



Object 2

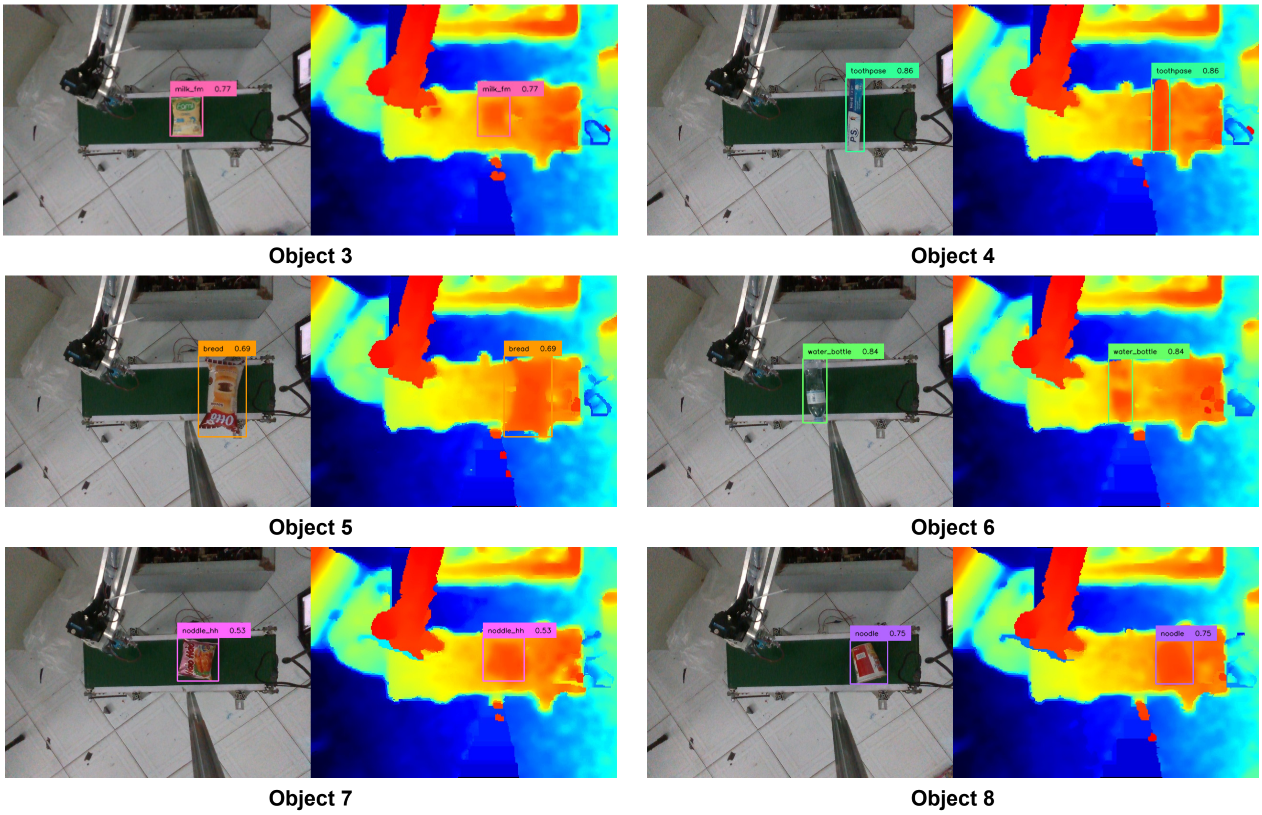


Figure 17. Detection results of RF-DETR on RGB and depth images showing multiple object types under different positions on the conveyor.

Figure 18 illustrates the real-world application of the robotic system, where a custom-designed two-finger parallel gripper is used to grasp objects detected on amoving conveyor belt. The robot is capable of hand–ling items with diverse physical properties, such as soft plastic packages, cylindrical cans, and rigid rectangular boxes. The successful manipulation of these objects validates the integration between the RF-DETR vision module and the robotic control system, confirming its effectiveness in precise grasping tasks. The combination of accurate visual perception and mechanical adaptabi–lity enables reliable operation in industrial palletizing and sorting environments.

To further assess the effectiveness of the RF-DETR-based robotic system, we conducted a total of 2,000 grasping trials across 10 object classes, using a two-finger parallel gripper integrated with real-time visual feedback. Each object was tested 200 times under varied orien–tations and placements on the conveyor system.As shown in Table 5, the system achieved near-perfect performance on most object classes, with several cate–gories such as *toothpaste*, *sausage*, *noodle_hh*, *milk_fm*, and *paper*reaching a 100% grasping success rate. These items are predominantly box-shaped, which aligns well with the geometry of the gripper and the planar surface of the conveyor. For objects with cylin–drical or rounded geometry, including *noodle*, *oolong*, *coca*, and *water_bottle*, the success rate ranged from 95% to 97.5%, indicating a slight decrease in reliability. Grasp failures in these cases were typically due to rotational slippage or marginal alignment between the contact surface and the gripper fingers. These experi–mental results confirm that the proposed system, which combines RF-DETR-based deep visual recognition with real-time 3D perception from an RGB-D camera, is capable of achieving reliable and adaptive object detec–tion and grasping in realistic industrial scenarios. The integration of transformer-based detection and depth-based geometric reconstruction enables consistent per–formance across diverse object types, poses, and occlusion levels.

Table 5. Grasp Performance of the Robotic System Across 10 Object Categories in a 2000-Trial Evaluation

Object Class	Total Attempts	Successful Grasps	Success Rate (%)
Toothpaste	200	200	100
Sausage	200	200	100
Noodle	200	190	95
Noodle_hh	200	200	100
oolong	200	194	97
Water_bottle	200	195	97.5
Coca	200	192	96
Bread	200	194	97
Milk_fm	200	200	100
Paper	200	200	100
Total	2000	1975	98.75

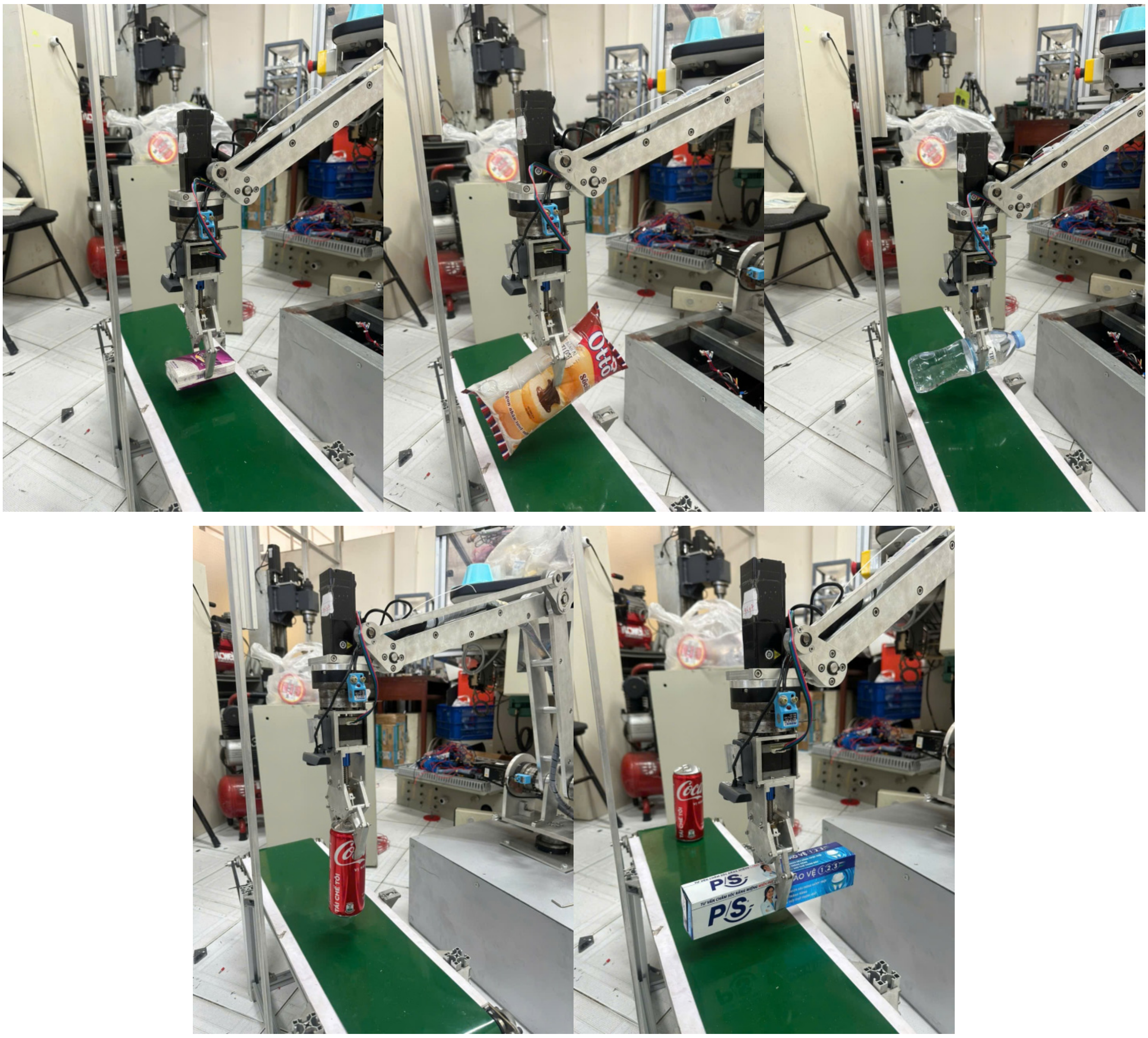


Figure 18. Robotic grasping execution using a two-finger parallel gripper guided by an RF-DETR vision system. Objects are successfully picked from the conveyor in real-world conditions.

6. CONCLUSION

This paper presented a complete development of a cost-effective palletizing robotic system that integrates mec–hanical design, embedded control, and deep-learning-based computer vision for industrial object handling. The study addresses the growing need for adaptable and low-cost automation solutions in retail and logistics envi–ronments, where variability in object shapes, materials, and arrangements presents persistent challenges.

The robotic platform was designed as a compact 4-degree-of-freedom manipulator, optimized for work–space coverage and mechanical stiffness while mainta–ining low production cost. The system utilizes standard AC servo motors, coupled with harmonic gear–boxes, for smooth and precise actuation. A Mitsubishi FX5U PLC serves as the low-level controller, ensuring high-speed signal processing and reliable motion execution. The modular architecture enables easy integ–ration and scalability for various industrial deployment scenarios.

To develop the vision system, an object detection module based on the RF-DETR transformer architecture was trained on a custom dataset of retail objects under varying lighting, background, and occlusion conditions. Combined with depth data from an Intel RealSense camera, the system estimates accurate 3D grasp poses from 2D detections through camera calibration and coordinate transformation. This fusion of 2D and 3D vision enables robust real-time performance in cluttered and semi-structured scenes.

Extensive experiments were conducted to validate the system under different palletizing conditions. The proposed setup achieved a grasping success rate of 98.75% across 2,000 trials, encompassing ten object categories, with high robustness to visual noise, object overlap, and challenging illumination conditions. Com–parisons with baseline detectors such as YOLOv12 and YOLOv8 further highlighted the superior performance of RF-DETR, particularly in scenarios with partial occlusion.

Future work will focus on expanding the system capabilities to handle dynamic object flows, integrating multi-view or articulated camera systems for occlusionaware detection, and applying reinforcement learning to enhance the adaptability of grasp strategies under real-world disturbances.

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Tahtawi, A. R., Hidayatullah, M. A., Cholid, M., 2021. Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics. Jour–nal of Robotics and Control (JRC), 2(6), 469–475.

[2] Vo, D. C., 2022. Design of a PLC-based control system for a SCARA robot arm in pick and place application. FME Transaction, 52(2), 226-236.

[3] Kim, H. S., and Kim, S. J., 2023. 5-Axis Robot Design for Loading and Unloading Workpieces. International Journal of Precision Engineering and Manufacturing, 24, 2279–2289.

[4] Xu, Z., Xia, J. and Zhong, F., 2024. Adaptive chic–ken swarm optimization algorithm for identifying structural parameters of 6-DOF mechanical arm. J Braz. Soc. Mech. Sci. Eng. 46(17). https://doi.org/10.1007/s40430-023-04585-0.

[5] Cheng, B., Wang, B., Chen, S., Huang, D., Dong, B., 2024. Investigation of axis-fitting-based measu–rement and identification techniques for kinematic parameters in multi-joint industrial robots. Preci–sion Engineering, 91, 1-13. https://doi.org/10.1016/j.precisioneng.2024.08.013.

[6] Wachter, A., Hartl-Nesic, C., Kugi, A., 2023. Robot Base Placement Optimization for Pick-and-Place Sequences in Industrial Environments. IFAC-PapersOnLine, 58(19), 19-24. https://doi.org/ 10.1016/j.ifacol.2024.09.080

[7] Meenakshi, P., et al. 2020. Remote controlled pick and place robot. IOP Conf. Series: Materials Science and Engineering, 1012, 1–8.

[8] Han, X., Liu, Y., Zhao, F., 2022. Design and deve–lopment of a compact desktop industrial robot without traditional reducers for low payload appli–cations. AES International Journal of Robotics and Automation (IJRA), 13(1), 31-40.

[9] Cong, V.D., 2023. Extraction and classification of moving objects in robot applications using GMM-based background subtraction and SVMs. J Braz. Soc. Mech. Sci. Eng. 45, 317. https://doi.org/10.1007/s40430-023-04234-6.

[10] Wei, H., Chen, X.X., Miao, X.Y., 2021. Vision-gui–ded fine-operation of robot and its application in eig–ht-puzzle game. Int J Int Robot Appl, 5(3), 576–589.

[11] Thotapalli, P.K., Vikram, K.C.R., Reddy, C.M., 2021. Feature extraction of moving objects using background subtraction technique for robotic applications. Int J Intell Robot Appl, 5, 65–78

[12] Nguyen, H.V., Cong, V.D., Trung, P.X., 2023. Development of a SCARA robot arm for palletizing applications based on computer vision. FME Transactions, 51, 541-549.

[13] Elassal, A., Abdelaal, M., Osama, M., Elhnidy, H., 2024. Low-cost parallel delta robot for a pick-and-place application with the support of the vision system. E-Prime - Advances in Electrical Engi–neering, Electronics and Energy, 8, 100518. https://doi.org/10.1016/j.prime.2024.100518

[14] Phuong, L.H, Cong, V.D, Hiep, T.T., 2023. Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision. FME Transactions, 51, 99-108

[15] Song, H.C., Song, J.B., 2013. Precision robotic de–burring based on force control for arbitrarily shaped workpiece using CAD model matching. Int. J. Precis. Eng. Manuf. 14, 85–91. https://doi.org/10.1007/s12541-013-0013-2

[16] Zheng, Z., Li, Y., Chen, D., Wang, L., Ju, J., Zhang, Q., 2023. Robot Target Location Based on the Difference in Monocular Vision Projection. IEEE Access, 11, 1883-1889.

[17] Singh, P., Rattan, M., Singh Grewal, N., Aggarwal, G., 2024. Distributed Feature Matching for Robust Object Localization in Robotic Manipulation. IEEE Access, 12, 161679-161687.

[18] Yang, H., Chen, L., Ma, Z., Chen, M., Zhong, Y., Deng, F., Li, M., 2021. Computer vision-based high-quality tea automatic plucking robot using Delta parallel manipulator. Computers and Elect–ronics in Agriculture, 181, 105946. https://doi.org/10.1016/j.compag.2020.105946

[19] Massah, J., Asefpour Vakilian, K., Shabanian, M., Shariatmadari, S. M., 2021. Design, development, and performance evaluation of a robot for yield estimation of kiwifruit. Computers and Electronics in Agriculture, 185, 106132. https://doi.org/10.1016/j.compag.2021.106132

[20] Zhang, C., Zhang, K., Ge, L., Zou, K., Wang, S., Zhang, J., Li, W., 2021. A method for organs clas–sification and fruit counting on pomegranate trees based on multi-features fusion and support vector machine by 3D point cloud. Scientia Horticulturae, 278, 109791. https://doi.org/10.1016/j.scienta.2020.109791

[21] Hussain M., 2023. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detec–tion. Machines. 11(7) https://doi.org/10.3390/machines11070677

[22] Diwan, T., Anirudh, G., Tembhurne, J.V., 2023. Object detection using YOLO: challenges, archi–tectural successors, datasets and applications. Mul–timed Tools Appl, 82, 9243–9275. https://doi.org/10.1007/s11042-022-13644-y

[23] Tan, L., et al., 2021. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. BMC Med Inform Decis Mak, 21, 324. https://doi.org/10.1186/s12911-021-01691-8

[24] Morera, Á., Sánchez, Á., Moreno, A.B., Sappa, Á.D., Vélez, J.F., 2020. SSD vs. YOLO for

Detection of Outdoor Urban Advertising Panels under Multiple Variabilities. Sensors, 20(16). https://doi.org/10.3390/s20164587

[25] Vilcapoma, P., Parra Meléndez, D., Fernández, A., Vásconez, I.N., Hillmann, N.C, Gatica, G., Vásconez, J.P., 2024. Comparison of Faster R-CNN, YOLO, and SSD for Third Molar Angle Detection in Dental Panoramic X-rays. Sensors, 24(18):6053. https://doi.org/10.3390/s24186053

[26] Jhang, J., Lin, C., 2024. Optimizing parameters of YOLO model through uniform experimental design for gripping tasks performed by an internet of things–based robotic arm. Internet of Things, 27, 101332. https://doi.org/10.1016/j.iot.2024.101332

[27] Panigrahi, S., Biyani, A., Mithari, A., Sawant, S., Tade, S., Kolte, M., 2023. YOLO on GPU Imple–mentation for a Vision-Based Fruit Select and Place System with a 5-DOF Robotic Arm, 7th International Conference On Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 1-7.

[28] Santos, A.A., Schreurs, C., da Silva, A.F. et al., 2024. Integration of Artificial Vision and Image Processing into a Pick and Place Collaborative Robotic System. J Intell Robot Syst 110, 159. https://doi.org/10.1007/s10846-024-02195-z

[29] Silveira, M., Santos, A.A., Pereira, F., da Ferreira Silva, A., FelgueirasRamosMachado, C.A.J., 2024. 3D Vision Object Identification Using Yolov8. Int. J. Mechatron. Appl. Mech. 17, 7–15.

[30] Li, Z., Xu, B., Wu, D., Zhao, K., Chen, S., Lu, M., Cong, J., 2023. A YOLO-GGCNN based grasping framework for mobile robots in unknown environ–ments. Expert Systems with Applications, 225, 119993. https://doi.org/10.1016/j.eswa.2023.119993

[31] Cheng-Jian, L., Jyun-Yu, J., Yi-Jyun, G., Hsiu-Mei, H., 2024. Vision-based Robotic Arm in Defect De–tection and Object Classification Applications, Sens. Mater., 36(2), 655-670.

[32] Hou, X., Liu, M., Zhang, S., Wei, P., Chen, B., Lan, X., 2025. Relation DETR: Exploring Explicit Posi–tion Relation Prior for Object Detection. Computer Vision – ECCV 2024. Lecture Notes in Computer Science, 15108. https://doi.org/10.1007/978-3-031-72973-7_6

[33] Huang, Y.X., Liu, H.I., Shuai, H.H., Cheng, W.H., 2025. DQ-DETR: DETR with Dynamic Query for Tiny Object Detection. In: Leonardis, Computer Vision – ECCV 2024. Lecture Notes in Computer Science, 15134. https://doi.org/10.1007/978-3-031-73116-7_17

[34] Hoanh, N., Pham, T. V., 2024. Focus-Attention Approach in Optimizing DETR for Object Detec–tion from High-Resolution Images. Knowledge-Based Systems, 296, 111939. https://doi.org/10.1016/j.knosys.2024.111939

[35] Li, H., Shi, F., 2024. A DETR-like detector-based semi-supervised object detection method for Brassica Chinensis growth monitoring. Computers and Electronics in Agriculture, 219, 108788. https://doi.org/10.1016/j.compag.2024.108788

[36] Peter, R., James, G., J, N., Isaac, R., 2025. RF-DETR: A SOTA Real-Time Object Detection Mo–del. Roboflow Blog: https://blog.roboflow.com/rf-detr/

ПРОЈЕКТОВАЊЕ И РАЗВОЈ РОБОТА ЗА ПАЛЕТИРАЊЕ СА ИНТЕЛИГЕНТНИМ 3Д СИСТЕМОМ ВИДА ЗАСНОВАНИМ НА РФ-ДЕТР МОДЕЛУ

Л.Х. Фуонг, Д.А. Дуј

Овај рад представља дизајн и имплементацију роботске руке за палетирање са 4 степена слободе (4-DOF) интегрисане са интелигентним 3Д системом вида заснованим на РФ-ДЕТР-у. Систем је усмерен на изазове детекције и манипулације објектима у неструктурираним индустријским окружењима која карактеришу неред, заклоњеност и варијабилност објеката. Роботска рука има лагану структуру високе крутости коју покрећу АЦ серво мотори и хармонијски мењачи, којима управља Митсубиши ФКС5У ПЛЦ. Обрада вида високог нивоа се врши на екстерном рачунару повезаном преко РС-485 протокола. РФ-ДЕТР модел је трениран на прилагођеном скупу података уобичајених малопродајних и логистичких објеката, укључујући кутије, боце и деформабилне пакете различитих облика, боја и материјала. Излази граничне кутије се пројектују у 3Д координате хватања користећи податке о дубини са Интел РеалСенс камере и конвертују се у координате робота путем калибрисаних трансформација. У поређењу са YOLOv12 и LW-DETR, предложени модел је постигао супериорне перформансе детекције (mAP@0,5:0,95 = 0,78), посебно под делимичном оклузијом, где YOLOv12 није успео. Систем је валидиран кроз 2.000 покушаја хватања у десет категорија објеката, постижући укупну стопу успеха од 98,75%. Сви правоугаони објекти су ухваћени са 100% тачношћу, док су се мање грешке јављале код цилиндричних предмета због клизања. Ови резултати показују робусност предложеног система у сложеним сценаријима и његову практичну одрживост за исплативо, вештачком интелигенцијом вођено палетизирање у реалним индустријским условима.