APPLICATION OF PLC IN CONTROLLING A DELTA ROBOT FOR COLOR-BASED PRODUCT SORTING

Dinh-Ngoc Le [1], Ngoc-Nhu-Buu Nguyen [2]*, Thanh-Tuan Huynh [3], Huy-Bao Nguyen [3], Duc-Dung Phan [3], Quang-Thuan Duong [3], Binh-Minh Tran [3], Van-Muon Nguyen [3]

[1] Phuc Nguyen Mechanical Electrical-Construction Company Limited
228/12/18, No. 8 Street, Linh Xuan Ward, HCMC, Vietnam

[2] Wembley Medical Factory Joint Stock Company
Lot I-10-7, D7 Street, Saigon Hi-Tech Park, Tang Nhon Phu Ward, HCMC, Vietnam

[3] Ho Chi Minh City University of Technology and Engineering (HCM-UTE)
Vo Van Ngan Str., No. 01, Ho Chi Minh City (HCMC), Vietnam
* Corresponding author. E-mail: buu.nnn@wembleymed.com.vn

Abstract: The application of PLC-based automation in product classification by criteria such as color is increasingly important in industrial production. However, direct experimentation in factories poses significant risks and costs. By miniaturizing the system and integrating a Delta robot with PLC S7-1200 and image processing, a laboratory platform was developed to simulate the real process. This model simplifies a complex industrial system into an experimental version that still ensures core functions such as sorting, accuracy, and efficiency. It serves as a valuable tool for in-depth research, teaching, and student practice in automation and robotics. Experimental results demonstrate the feasibility of the system and its potential for further improvement toward industrial applications.

Keywords: PLC S7-1200, Delta Robot, Image Processing, Automation.

1. Introduction

In the context of Industry 4.0, the demand for automation and intelligent production lines has been growing rapidly. Industrial robots, with their capability to perform repetitive tasks at high speed and precision, play a central role in this transformation. Among different robot architectures, the Delta robot, invented by Professor Reymond Clavel at EPFL in the 1980s, is recognized for its parallel structure with three lightweight arms connecting a fixed base to a moving platform. Thanks to this design, Delta robots achieve outstanding dynamic performance, offering both high speed and good accuracy. They have been widely applied in packaging, assembly, product classification, and other operations where efficiency and reliability are critical [1].

Despite its potential, current research and applications of Delta robots in sorting tasks still reveal many limitations. A large proportion of existing works focus on basic conveyor-cylinder sorting systems [2][3]. These mechanisms are attractive due to their simplicity and low cost, but they inherently lack flexibility and cannot satisfy the increasing requirements for high throughput and precision demanded by modern manufacturing.

Other studies attempt to employ basic robotic arms controlled by microcontrollers or PCs [4]. Such systems provide a convenient platform for academic research and laboratory experiments, but they are not robust enough for industrial deployment. They usually suffer from latency, limited real-time performance, and insufficient reliability under harsh factory environments.

Several efforts have also introduced low-cost Delta robot prototypes using 3D printing and common components [5]. These works successfully demonstrate the feasibility and accessibility of Delta robots, especially for education and experimental purposes. However, the scope of these prototypes is mostly confined to fundamental pick-and-place functionality, without considering advanced features such as real-time image processing, industrial reliability, or integration into automated production lines.

More recently, some works have explored the application of PLCs in sorting systems [6][7]. These studies confirm that PLCs provide high stability, ease of integration, and compatibility with industrial standards. However, the combination of PLC motion control, Delta robot kinematics, and real-time vision processing has not been fully exploited.

To address these limitations, this paper proposes a fully integrated Delta PLC-robot-vision platform. The system explicitly integrates the mathematical model of the Delta robot, including forward kinematics and inverse kinematics, and uses real-time image processing to extract the shape and color of the object. The Siemens S7-1200 PLC performs deterministic motion control by generating synchronous pulse train (PTO) outputs, while the TCP/IP communication interface transmits the kinematic solutions from the computer to the PLC to realize the precise trajectory. The proposed architecture bridges the gap between theoretical modeling and industrial-level implementation, providing a reliable and scalable method for multi-criteria product sorting.

This paper addresses that gap by developing a PLC-Delta robot-real-time image processing integrated platform for product sorting. The proposed system combines the industrial reliability of PLC, the high speed and accuracy of Delta robots, and the flexibility of modern image processing technology, offering a practical solution for automated classification in industrial environments. This contribution not only enhances system performance but also demonstrates a new direction for applying Delta robots in smart factories.

2. Model

The Delta robot is a three-DOF translational parallel mechanism composed of three identical kinematic chains. Each chain consists of an upper active arm of length r_f, driven by a rotary motor at the fixed base, and a lower passive parallelogram of length r_e, ensuring pure translational motion of the mobile platform.

2.1. Mathematical Model

2.1.1. Forward Kinematics

The mathematical model of the Delta robot is established based on the geometric and kinematic principles of the three-degree-of-freedom parallel mechanism. The robot structure consists of three active arms connected to the fixed table and three passive arms in the form of parallelograms connected to the mobile table, ensuring the translational motion of the working head.

The forward kinematic problem determines the coordinates of the operating point (x, y, z) when knowing the three rotation angles of the delta robot (θ_1, θ_2, θ_3) through the spherical equation system:

$$\begin{gathered}(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_e{}^2 \\ i = 1, 2, 3\end{gathered} \tag{1}$$

Where (x_i, y_i, z_i) are the coordinates of the elbow joints, which depend on the input angles θ_i.

On the contrary, the inverse kinematic problem determines the joint angles from the desired position of the mobile table, based on the geometric relationship between the links in the plane of motion (YZ) and the symmetry of the mechanism. This model is the basis for simulation and design of Delta robot control algorithm in applications requiring high precision.

2.1.2. Inverse Kinematics

Given a target position (x, y, z), the required actuator angles θ_i are determined by solving the intersection between the passive-arm sphere and the active-arm circular rotation plane. For each arm i, the angle θ_i is calculated using the following auxiliary parameters:

$$E_i = 2r_f(y_i - y_{0i}) \tag{2}$$

$$F_i = 2r_f(z_i - z_{0i}) \tag{3}$$

$$\begin{gathered}G_i = x_i^2 + y_i^2 + z_i^2 + r_f^2 - r_e^2 + y_{0i}^2 + z_{0i}^2 \\ -2(x_i x_{0i} + y_i y_{0i} + z_i z_{0i})\end{gathered} \tag{4}$$

Where (x_{0i}, y_{0i}, z_{0i}) are the coordinates of the motor joint on the fixed base. The joint angle θ_i is then found by:

$$\theta_i = 2\,arctan\left(\frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i}\right) \tag{5}$$

This analytical form ensures efficient real-time computation suitable for high-speed sorting.

2.2. Hardware Structure

The hardware system includes the following main components:

- Central processing block: PLC Siemens S7-1200 (1214 DC/DC/DC) combined with expansion module SM1222DQ, responsible for signal processing and control [8].
- Drive system: 2-phase stepper motor FY56EC350A controlled through driver DM542, helping the Delta robot arm operate accurately. Sensor and relay: Proximity sensor LJ12A3-4-Z/AY is used to detect objects, intermediate relay performs on/off control signal.
- Data acquisition block: USB camera connected to computer, takes product photos and recognizes colors through EmguCV library in C# [9].
- Actuator: Conveyor belt to transport products and vacuum pump for Delta robot to perform pick-up and drop operations.

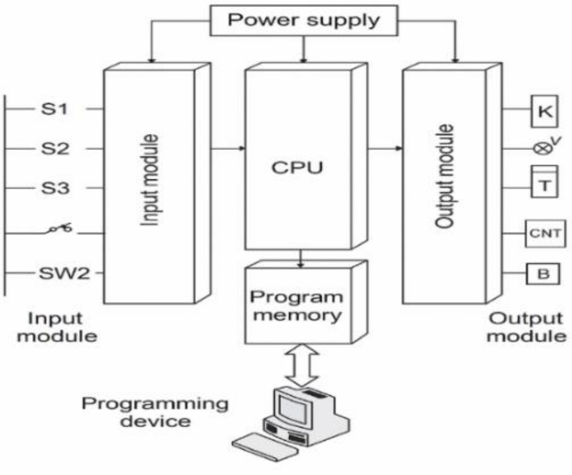


Fig. 1. PLC control system.

2.3. Working Principle

The products are transported on the conveyor belt and pass through the camera. The captured images will be processed in real time by C# software to recognize colors.

The recognition results are transmitted to the PLC via Ethernet protocol. The PLC processes this signal and issues control commands to the output modules, thereby controlling the stepper motors and actuators. The Delta robot performs the picking and sorting of products into appropriate locations according to color.

2.4. Control and Communication Architecture

The platform adopts a Master-Slave configuration.

2.4.1. Master Side (PC Application)

A C# application performs the following tasks:

- HSV-based color segmentation.
- Morphological filtering.
- Contour extraction and polygonal approximation.
- Centroid computation using image moments:

$$C_x = \frac{M_{10}}{M_{00}}, C_y = \frac{M_{01}}{M_{00}} \tag{6}$$

- Pixel-to-workspace mapping to convert image coordinates to robot physical coordinates..
- Inverse kinematic computation of θ_1, θ_2, θ_3.

The PC then transmits a compact command packet - including joint angles, velocity, and action flags - to the PLC via TCP/IP.

2.4.2 Slave Side (PLC Execution)

The PLC processes the received data and generates synchronized PTO pulses for coordinated three-axis motion. A finite state machine (FSM) governs system behavior: homing, approach, pick, transfer, release, and reset. Acceleration/deceleration ramps are applied to ensure smooth motion and reduce mechanical stress.

2.5. Vision Processing Pipeline

The vision module follows a six-step pipeline:

1. RGB-to-HSV conversion for illumination robustness.
2. Color thresholding to detect red, yellow, and blue objects.
3. Morphological opening/closing to remove noise and refine object boundaries.
4. Contour detection and polygon fitting, where object categories are determined from vertex.
5. Centroid extraction for position determination.
6. Coordinate transformation to robot workspace, used as input to the inverse kinematics routine.

The combination of deterministic robot motion (PLC) and flexible perception (PC) enables accurate, real-time multi-criteria sorting.

2.6 Software design

- PLC program: Programmed on TIA Portal with Ladder Logic language to handle control logic.
- Image processing: Developed in C# integrating EmguCV, helping to identify product colors in real time.
- Database: SQL Server is used to store classification data, support monitoring and analysis later.

3. Control Method

3.1. System Control Architecture

The system employs a Master-Slave control architecture to leverage the strengths of both PC-based processing and PLC-based industrial control.

The Master (PC): Handles computationally intensive tasks that are non-deterministic, such as Image Processing (EmguCV) and Inverse Kinematics calculations. A standard PC cannot guarantee the precise microsecond-level timing required for smooth stepper motor control.

The Slave (PLC S7-1200): Acts as the Real-time Motion Controller. The PLC is essential for this setup because:

- Deterministic Pulse Generation: It utilizes High-Speed Output (PTO) channels (up to 100 kHz) to drive three stepper motors simultaneously with precise synchronization, which a PC operating system cannot achieve directly.
- Industrial Reliability: The PLC handles safety interlocks (Emergency Stop, Limit Switches) and ensures the system stops safely even if the PC software crashes.
- Signal Conversion: It acts as a robust interface between the 5V logic of the PC/Sensors and the 24V industrial standard of the actuators.

Communication Protocol: The PC sends a data packet {Command, θ_1, θ_2, θ_3, Speed} to the PLC via TCP/IP. The PLC parses this packet and executes the motion profile using its internal Pulse Train Output (PTO) instructions.

3.2. Vision-Based Recognition Algorithm

The image processing procedure (Fig. 2) is designed to be robust against industrial lighting variations. The core algorithm consists of the following steps:

Preprocessing: The raw image from the camera is converted from RGB to the HSV color space (Hue, Saturation, Value). Using HSV decouples chromatic information (Hue) from lighting intensity (Value), significantly improving color detection stability under fluctuating ambient light. The HSV thresholds were empirically tuned: Red ($H \in [0, 10]$), Blue ($H \in [100, 120]$), and Yellow ($H \in [20, 30]$).

Segmentation & Morphology: A thresholding technique is applied to the HSV channels to generate a binary image. Subsequently, morphological operations-specifically Dilation and Erosion-are performed iteratively to eliminate "salt-and-pepper" noise and close any discontinuities in the object's contour.

Feature Extraction:

- Centroid calculation: The precise center (C_x, C_y) of the object is computed using Image Moments (M_{ij}), ensuring accurate coordinate localization for the robot's end-effector.

- Shape classification: The Contour Approximation algorithm (based on the Douglas-Peucker method) is applied to simplify the object's boundary. The system classifies the shape based on the number of vertices detected (e.g., 3 vertices for a Triangle, 4 for a Rectangle).

Coordinate Calibration: To map the pixel coordinates (u, v) from the image to the robot's physical coordinate system (x, y), a Coordinate Calibration process was performed. We used a linear transformation model:

$$x_w = K_x(u - u_0) + x_{offset} \quad (7)$$

$$y_w = K_y(v - v_0) + y_{offset} \quad (8)$$

Where K_x, K_y are scaling factors (mm/pixel) determined experimentally. In our setup, with the camera mounted at a height of 400mm, the resolution is 640 × 480, resulting in a scaling factor of approximately 0.56 mm/pixel.

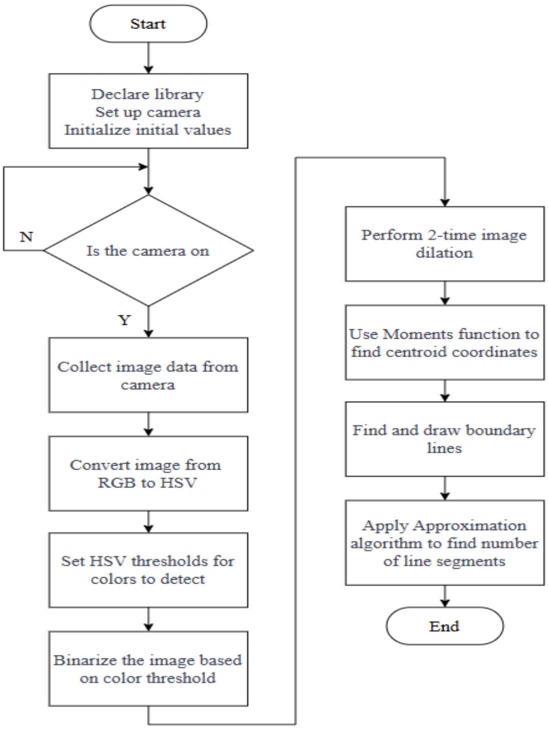


Fig. 2. Image processing flowchart.

3.3. Robot Control Algorithm

The control logic (Fig. 3) follows a Finite State Machine (FSM) principle to ensure logical sequential execution:

- Initialization: The system verifies the connection status of the PLC, Camera, and safety sensors. The robot performs a homing sequence to establish the machine coordinate system.
- Detection & Coordinate Transformation: Upon object detection, the object's pixel coordinates (u, v) from the image are transformed into the robot's physical world coordinates (x_w, y_w) using a calibration matrix.
- Trajectory Planning: The PC calculates the Inverse Kinematics to determine the three required joint angles (θ_1, θ_2, θ_3). These values are converted into pulse numbers and transmitted to the PLC.
- Sorting Execution: Based on the identified attributes (Color + Shape), the PLC executes the pick-and-place trajectory to sort the item into the designated bin (e.g., Red Triangle → Bin 1, Blue Rectangle → Bin 2).

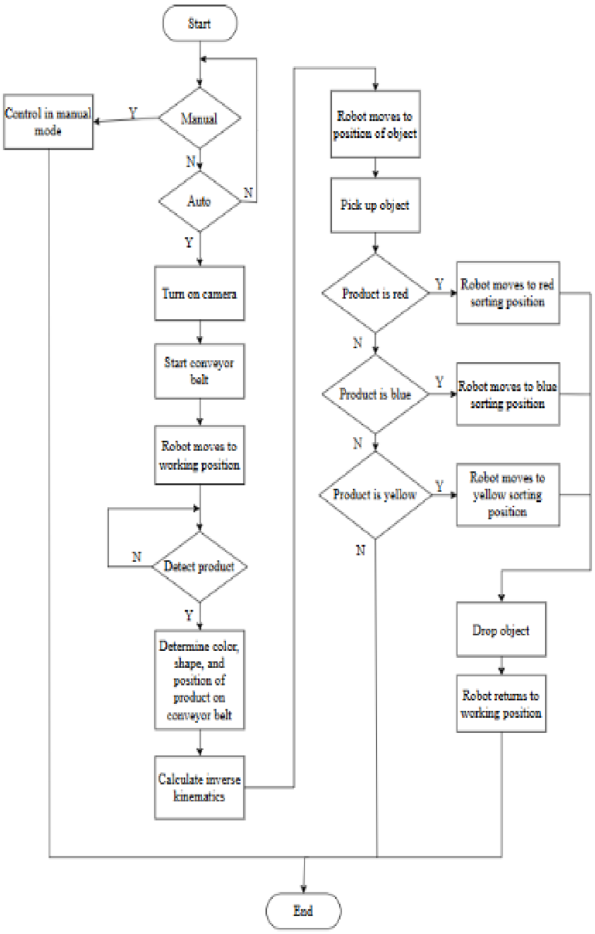


Fig. 3. Robot control flowchart.

4. Experiments

4.1. Real Mode

The experiments were performed on the custom-built sorting platform, shown in Fig. 4. This platform is composed of two primary assemblies: the Control Panel (top), which houses the Siemens S7-1200 PLC, motor drivers, and power supply; and the Delta Robot System (bottom), which includes the aluminum fixed frame, the parallel robot arms, and the conveyor belt.

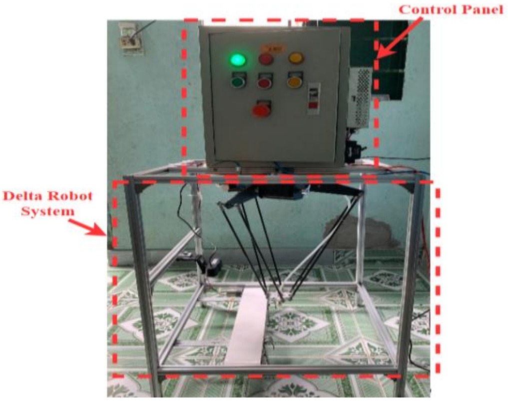
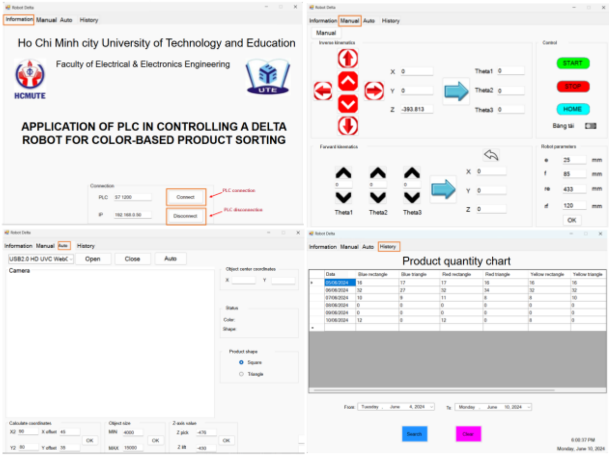


Fig. 4. The physical experimental platform.

The system is operated via a custom Graphical User Interface (GUI) developed in C#, as shown in Fig. 5. This interface allows the operator to establish or terminate the connection to the PLC ("Connect"/"Disconnect"), select operational modes ("Manual", "Auto"), and access historical sorting data ("History").



Ho Chi Minh city University of Technology and Education

Faculty of Electrical & Electronics Engineering

APPLICATION OF PLC IN CONTROLLING A DELTA ROBOT FOR COLOR-BASED PRODUCT SORTING

Connection
PLC: S7 1200
IP: 192.168.0.50

X: 0
Y: 0
Z: -393.813
Theta1: 0
Theta2: 0
Theta3: 0

e: 25 mm
f: 85 mm
re: 433 mm
rf: 120 mm

USB2.0 HD UVC WebC

Color:
Shape:

● Square
○ Triangle

X2: 90 X offset: 45
Y2: 80 Y offset: 38
MIN: 4000
MAX: 15000
Z pick: -476
Z lift: -430

Product quantity chart

Date	Blue rectangle	Blue triangle	Red rectangle	Red triangle	Yellow rectangle	Yellow triangle
05/06/2024	16	17	17	16	16	16
06/06/2024	32	27	32	34	32	32
07/06/2024	10	9	11	8	8	10
08/06/2024	0	0	0	0	0	0
09/06/2024	0	0	0	0	0	0
10/06/2024	12	0	12	0	8	0

From: Tuesday, June 4, 2024 To: Monday, June 10, 2024

6:00:37 PM
Monday, June 10, 2024

Fig. 5. The main control interface.

4.2. Experimental Result

The validation of the machine vision algorithm was a critical part of the experiment. The system was tested against objects of different colors and geometric shapes.

Actual results of the image recognition process are presented in Fig. 6 and Fig. 7:

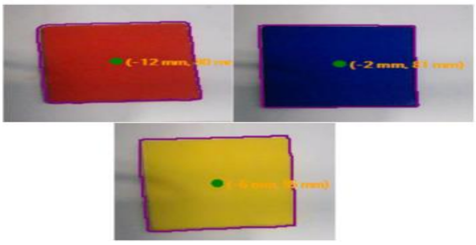


Fig. 6. Results of image processing for rectangular objects.

The image recognition procedure successfully segmented the object boundaries and identified their respective colors (Red, Blue, and Yellow). Critically, the system accurately calculated and displayed the centroid coordinates (x, y) in millimeters for each object. For instance, the red object was localized at (−12 mm, 80 mm), and the yellow object at (−6 mm, 85 mm). These coordinates represent the precise pick-up points required for the subsequent robot control actions.

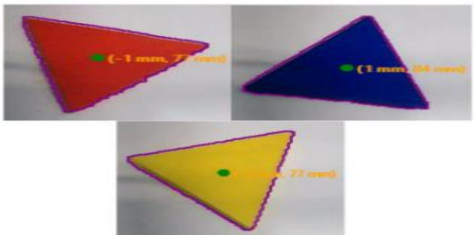
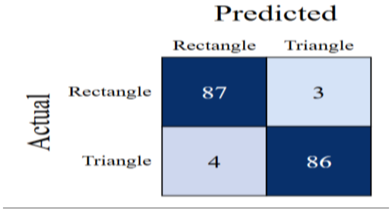


Fig. 7. Results of image processing for triangular objects.

The system maintained high fidelity across different shapes, correctly classifying the objects as triangles and recognizing their colors. Similar to the rectangular objects, the system provided the precise centroid coordinates: the red triangle was localized at (−1 mm, 77 mm), and the blue triangle at (1 mm, 84 mm). The consistency in localization across different shapes and colors confirms the robustness of the image processing algorithm. The calculated coordinates provide the essential real-time positional data needed by the PLC for execution of the color-based product sorting task.

To validate the performance of the integrated PLC-Delta Robot system using Computer Vision, an experiment was conducted with a total of 180 product samples. The dataset was balanced across two geometric shapes (Rectangle, Triangle) and three colors (Red, Blue, Yellow), with 30 samples for each category. The experimental results are summarized in



	Rectangle	Triangle
Rectangle	87	3
Triangle	4	86

Fig. 8. Confusion matrix for geometric shape classification performance.

Correct Classifications: 87 actual Rectangles were correctly identified as Rectangles, and 86 actual Triangles were correctly identified as Triangles. The total number of correctly sorted items is 87 + 86 = 173.

Misclassifications: Only 3 actual Rectangle samples were incorrectly classified as Triangles, and 4 actual Triangle samples were incorrectly classified as Rectangles. The total number of misclassified items is 3 + 4 = 7.

Performance Analysis: The system successfully classified 173 out of 180 samples, achieving an overall accuracy of 96.11%. The breakdown of the results is as follows:

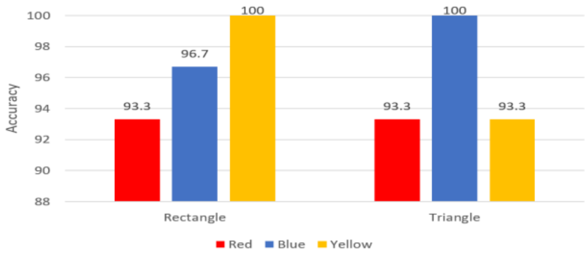


Fig. 9. Performance metrics comparison.

Rectangular Products: The system achieved a high average accuracy of 96.67%. Notably, the Yellow Rectangles were identified with 100% precision. Minor misclassifications occurred in the Red and Blue categories (93.3% and 96.7% accuracy, respectively).

Triangular Products: The average accuracy was 95.56%. The system performed perfectly for Blue Triangles (100% accuracy). A slightly lower accuracy of 93.3% was observed for Red and Yellow triangles.

Conclusion: The experimental data confirms the robustness of the proposed system. The classification errors were minimal and primarily attributed to external lighting fluctuations affecting color thresholding or minor edge approximation jitters for triangular shapes. With the lowest accuracy in any category remaining above 93%, the system demonstrates high feasibility for industrial sorting applications.

5. Conclusions

This study successfully implemented a vision-based Delta robot platform utilizing a hierarchical PC-PLC Master-Slave architecture. Experimental results on 180 samples demonstrated a classification accuracy of 96.11%. These findings validate the system's feasibility for automated industrial sorting and its value as an experimental educational platform. Future work will explore Deep Learning integration to enhance object recognition capabilities.

Acknowledgement

We want to give thanks to Assoc. Prof. Minh-Tam Nguyen (HCM-UTE) due to his supervision for this study. The video link of operation of systems is: https://www.youtube.com/watch?v=m7wQIKzuPcU&feature=youtu.be

6. References

[1] Bonev I.: "Delta parallel robot—the story of success", Parallel Mechanisms Information Center, 2001. Available: http://www.parallemic.org/Reviews/Review002.html

[2] Elghitany M.N.: "A comprehensive review of Delta robot research", International Journal of Robotics and Automation, vol. 39, no. 2, pp. 87–102, 2024.

[3] Elassal A. et al: "Low-cost parallel Delta robot for pick-and-place application", in Proc. 12th Int. Conf. on Mechatronics and Robotics Engineering, pp. 112–118, 2024.

[4] Sanwar S. et al: "Automated object sorting system with real-time image processing", Journal of Automation and Control Engineering, vol. 11, no. 3, pp. 45–53, 2023.

[5] Guerra M.: "Storage system for automated object classification based on color under uncontrolled environment", Sensors and Applications, vol. 18, no. 6, pp. 235–244, 2024.

[6] Awoda M. And Ali R.: "PLC-based implementation of automated sorting systems", in Proc. TIJER Conf. 2023, pp. 210–215, 2023.

[7] Ali R. and Awoda M.: "Industrial automation using PLC in color sorting applications", Int. J. of Industrial Electronics and Automation, vol. 15, no. 4, pp. 301–310, 2024.

[8] Siemens AG: SIMATIC S7-1200 Programmable Controller System Manual, 2020.

[9] Shakhnarovich G. , Darrell, T. and Indyk P.: "Nearest-Neighbor Methods in Learning and Vision", Theory and Practice, MIT Press, 2006.

[10] Williams R.L.: "The Delta Parallel Robot: Kinematics Solutions", Ohio University Mechanical Engineering, 2016.

[11] IEEE NITK: "OpenCV: Computer Vision Library", IEEE Student Branch NITK. Available: https://ieee-nitk.github.io/blog/opencv/

[12] Sharma A. et al: "Object Detection using OpenCV and Python", 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), India, pp. 501-505, 2021.

[13] Vo D.H. et al: "Position Control of 3-DOF Experimental Articulated Robot Arm using PID Controller", JFSC, vol. 3, no. 1, pp. 73–80, Mar. 2025.

[14] Garetti, M. et a: „Life Cycle Simulation for the design of Product Service Systems, Computers in Industry", Vol. 63, pp. 361-369, 2012.