Journal of Fuzzy Systems and Control, Vol. 4, No 3, 2026
ISSN: 2986-6537, DOI: 10.59247/jfsc.v4i3.387

Sorting Model using Robotic Arm with Image Processing

Nguyen-Khoa Tran [1], Dinh-Khang Nguyen [2], Phong-Luu Nguyen [3,*], Nhat-Anh Huynh [4], Thanh-Hung Tran [5], Khac-Dinh Nguyen [6], Xuan-Anh Dinh [7], Binh-Hau Nguyen [8], Gia-Phu Nguyen [9], Minh-Phuoc Cu [10]
[1, 2, 3, 4, 7, 9] Ho Chi Minh City University of Technology and Engineering (HCM-UTE), Ho Chi Minh City (HCMC), Vietnam
[5, 8] Posts and Telecommunication Institute of Technology (PTIT), Ho Chi Minh City (HCMC), Vietnam
[6] Hyosung Dong Nai Company, Dong Nai Province, Vietnam
[10] Cao Thang Technical College, Ho Chi Minh City (HCMC), Vietnam
Email: [1] 22851009@student.hcmute.edu.vn, [2] 22851007@student.hcmute.edu.vn, [3] luunp@hcmute.edu.vn, [4] 22144049@student.hcmute.edu.vn, [5] hungtt@ptit.edu.vn, [6] khacdinh07054@gmail.com, [7] 23151049@student.hcmute.edu.vn, [8] haunb@ptit.edu.vn, [9] 23151157@student.hcmute.edu.vn, [10] cuminhphuoc@caothang.edu.vn
*Corresponding Author

***Abstract*—This paper presents the design and implementation of a product sorting model using a robotic arm integrated with image processing techniques. The system consists of a conveyor belt, a vision module, and robotic manipulators that work together to identify and classify objects through a camera and computer vision algorithms that detect product characteristics. The robotic arm then performs the corresponding sorting operation according to product quality requirements. The hardware design includes the construction of the robotic arm, control circuits, and integration with actuators, while the software design focuses on developing image processing algorithms and communication between the vision system and the robot controller. Experimental results show that the system achieves an average size measurement error of approximately ±2 mm, a classification accuracy of about 95%, and an average processing time of 2–3 seconds per product. These results demonstrate reliable recognition and classification performance compared to some previous research models. The proposed model emphasizes the feasibility of combining robotic manipulation and computer vision for automated sorting tasks in industrial applications such as food processing, household tools, and medical instruments, while also serving as a practical training platform for students in technical education. Future improvements may include optimizing vision algorithms, enhancing the mechanical design of the robotic arm, integrating artificial intelligence to improve safety, and expanding the system's capability to handle more complex classification tasks.**

Keywords*—Automation; Computer Vision; Image Processing; Industrial Applications; Object Sorting; Robotic Arm

I. INTRODUCTION

Automated object classification using robotic arms has become an important approach in modern manufacturing, where accuracy, repeatability, and operational flexibility are required. Traditional manual sorting is labor-intensive and may be inconsistent for repetitive, high-throughput tasks. Early studies [1]-[4], demonstrated the feasibility of color- and dimension-based sorting with image processing and robotic systems, while related computer-vision research has also addressed visual alignment and measurement tasks [5].

Subsequent research improved detection reliability by applying HSV-based color segmentation and contour-based shape recognition [6]. Other studies investigated low-cost OpenCV-based robotic systems [7], multi-feature classification using sensors and embedded platforms [8], and camera-based image-processing systems using platforms such as Raspberry Pi [9]-[11]. Camera calibration and vision-guided robot coordination have also been investigated to improve positioning accuracy in robotic integration [12]-[15]. Recent vision-guided robotic research has increasingly incorporated learning-based perception and explicit robot-camera calibration for grasping and localization [16]-[18].

Nevertheless, many existing solutions remain focused on laboratory-scale implementations and may face limitations related to real-time performance, dimensional accuracy, or integration with industrial controllers. To address these issues, the present research develops an integrated system combining OpenCV-based computer vision, a PLC-based motion-control architecture, a three-joint robotic arm, and an HMI interface. The system performs object detection, dimension measurement, coordinate transformation, and size-based robotic sorting. Unlike recent learning-based approaches [16]-[18], the present study deliberately uses a deterministic OpenCV pipeline to provide a low-cost and computationally simple laboratory platform. The proposed model is intended primarily for laboratory and small-scale production applications; therefore, performance under diverse industrial conditions remains a limitation of the present study.

II. MODEL

This section presents the kinematic formulation and physical design of the three-degree-of-freedom robotic arm. The robot specification is listed in Table 1.

Table 1. Physical Specifications of The Model

Parameter	Meaning
L_1=240mm	Length of link 1
L_2=177mm	Length of link 2
L_3=134mm	Length of link 3
d_1=270mm	Base-to-Joint 1 offset
d_3=55mm	Joint 3 offset

A. Kinematic

1. Forward Kinematic

The forward and inverse kinematics are formulated using the Denavit-Hartenberg (DH) representation of the robot

(Fig. 1). The Denavit–Hartenberg parameters for the three-joint robotic arm are listed in Table 2.

Using the D-H convention in Table 2, the homogeneous transformation matrix ${}^{i-1}_{i}T$:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\alpha \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

$$ {}^{0}_{3}T = {}^{0}_{1}T.{}^{1}_{2}T.{}^{2}_{3}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

$$ P_x = L_1\cos\theta_1 + d_3\sin\theta_1 + L_2\cos\theta_1\cos\theta_2 + L_3\cos\theta_1\cos\theta_2\cos\theta_3 - L_3\cos\theta_1\sin\theta_2\sin\theta_3 \tag{3} $$

$$ P_y = L_1\sin\theta_1 - d_3\cos\theta_1 + L_2\cos\theta_2\sin\theta_1 + L_3\cos\theta_2\cos\theta_3\sin\theta_1 - L_3\sin\theta_1\sin\theta_2\sin\theta_3 \tag{4} $$

$$ P_z = d_1 + L_2\sin(\theta_2+\theta_3) + L_2\sin(\theta_2) \tag{5} $$

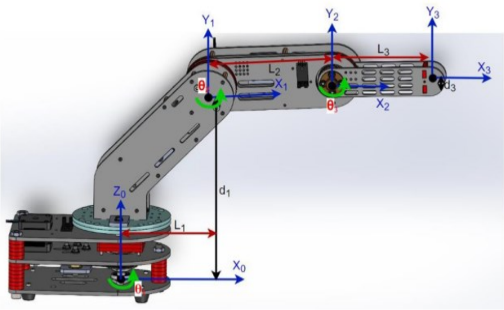

Fig. 1. Coordinate axes for the robot arm system

Table 2. DH Table

i	a_i	α_i	d_i	θ_i
1	L_1	90°	d_1	θ_1
2	L_2	0	0	θ_2
3	L_3	0	d_3	θ_3

2. *Inverse Kinematic*

The inverse-kinematic solution is obtained by rearranging the transformation relationship and comparing the corresponding matrix elements. To derive the inverse kinematic equations, we premultiply both sides of (2) by ${}^{1}_{0}T^{-1}$, yields:

$$ {}^{1}_{0}T^{-1}.{}^{3}_{0}T = {}^{1}_{0}T^{-1}.({}^{1}_{0}T.{}^{2}_{1}T.{}^{3}_{2}T) \tag{6} $$

By evaluating the elements of the corresponding transformation matrices on both sides, the position components are mapped into intermediate expressions as shown in (7):

$$ \begin{bmatrix} * & * & * & n_1 \\ * & * & * & n_2 \\ * & * & * & n_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} * & * & * & m_1 \\ * & * & * & m_2 \\ * & * & * & m_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

Equating the position vectors from both sides gives the set of simplified (8):

$$ \begin{cases} n_1 = P_x\cos\theta_1 - L_1 + P_y\sin\theta_1 \\ n_2 = P_z - d_1 \\ n_3 = P_x\sin\theta_1 - P_y\cos\theta_1 \\ m_1 = L_3\cos(\theta_2+\theta_3) + L_2\cos\theta_2 \\ m_2 = L_3\sin(\theta_2+\theta_3) + L_2\sin\theta_2 \\ m_3 = d_3 \end{cases} \tag{8} $$

From the matrix identity in (7), the third row of the position vector implies $n_3 = m_3$. Substituting the expressions from (8) into this equality leads to the trigonometric equation for θ_1:

$$ n_3 = m_3 \Rightarrow P_x\sin\theta_1 - P_y\cos\theta_1 = d_3 \tag{9} $$

To solve (9), we apply the auxiliary angle technique. By dividing both sides by $\sqrt{P_x^2+P_y^2}$ and introducing an auxiliary angle α, the equation can be rewritten using the sine addition formula in (10):

$$ \begin{cases} \cos\alpha = \frac{P_x}{\sqrt{P_x^2+P_y^2}};\ \sin\alpha = \frac{-P_y}{\sqrt{P_x^2+P_y^2}};\ \alpha\in[0,2\pi] \\ \sin\theta_1\cos\alpha + \cos\theta_2\sin\alpha = \frac{d_3}{\sqrt{P_x^2+P_y^2}} \\ \theta_1 = \arcsin\left(\frac{d_3}{\sqrt{P_x^2+P_y^2}}\right) - \alpha \end{cases} \tag{10} $$

Where the auxiliary angle α is determined based on the signs of P_x and P_y as described in (11):

$$ \begin{cases} \alpha = \arcsin\left(\frac{-P_y}{\sqrt{P_x^2+P_y^2}}\right); (P_x > 0, P_y > 0) \\ \alpha = \arccos\left(\frac{-P_x}{\sqrt{P_x^2+P_y^2}}\right); (P_x < 0, P_y < 0) \end{cases} \tag{11} $$

Once θ_1 is obtained, the remaining joint variables θ_2 and θ_3 can be calculated.

Let:

$$ \theta_{23} = \theta_2 + \theta_3 $$

Represent the combined angle of links 2 and 3. Substituting θ_{23} into the P_z position equation (5), we obtain (12).

$$ \begin{aligned} P_z &= d_1 + L_2\sin(\theta) + L_2\sin(\theta_2) \\ \Rightarrow \theta_2 &= \arcsin\left(\frac{P_z - d_1 - L_3\sin\theta_{23}}{L_2}\right) \end{aligned} \tag{12} $$

The final joint variable is obtained in (13) from the relationship between the total end-effector angle and the two planar joint angles.

$$ \Rightarrow \theta_3 = \theta_{23} - \arcsin\left(\frac{P_z - d_1 - L_3\sin\theta_{23}}{L_2}\right) \tag{13} $$

B. Model Design

The robotic arm is designed to pick and place workpieces from the conveyor to the sorting area while providing sufficient workspace coverage, positional accuracy, and stable operation. The mechanism has three Degrees of Freedom (3 DOF): a base joint for horizontal rotation, a shoulder joint for vertical movement, and an elbow joint for adjusting the reach of the end-effector. The projected views and three-dimensional representation of the robotic arm are shown in Fig. 2 and Fig. 3.

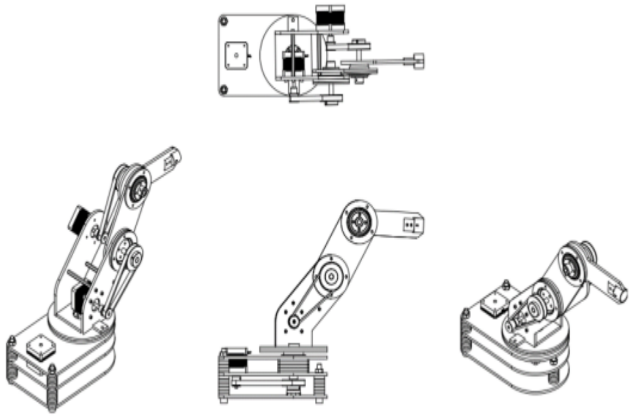

Fig. 2. Robot projections

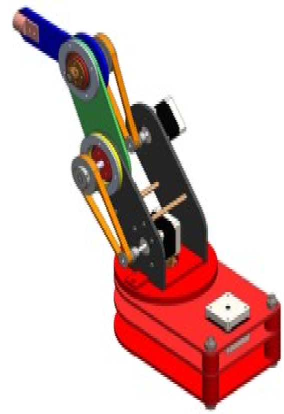

Fig. 3. Three-dimensional representation of the robotic arm

The structural components are manufactured from lightweight mica to reduce moving mass while maintaining adequate rigidity. Sliding bearings are used at the joints to reduce friction, and the joints are actuated by stepper motors through gear or belt transmission. This arrangement is intended to support stable motion and positioning of the end-effector.

The control system uses one workpiece-presence sensor, three limit switches, and one ON push button as input signals. The outputs include six pulse-and-direction signals for the three robot axes, together with control signals for the conveyor motor, vacuum pump, and suction lifting actuator. Because the controller must generate high-speed pulses for the stepper motors, the Mitsubishi FX3U-24MT PLC is selected for the proposed system. The selected PLC and motion-control hardware are shown together in Fig. 4.

The PLC wiring and actuator connections are arranged to provide the required input and output interfaces for the robot axes, conveyor, sensors, and suction subsystem. The corresponding wiring arrangement is shown in Fig. 5.

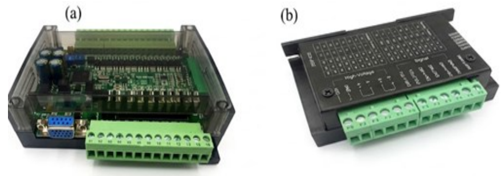

Fig. 4. PLC and motion-control hardware: (a) Mitsubishi FX3U-24MT PLC and (b) TB6600 stepper driver

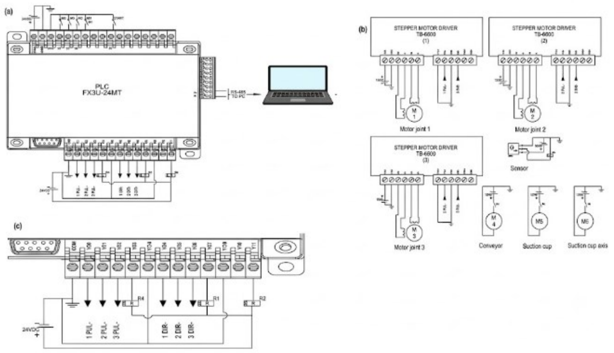

Fig. 5. PLC wiring and actuator connection diagram

The main mechanical and actuation components used in the prototype are shown together in Fig. 6. The transmission components include the GT2 pulleys and timing belt, while the actuation components include the robot stepper motor, conveyor motor, vacuum pump, and suction actuator.

Fig. 6. Main mechanical and actuation components used in the prototype: (a) GT2 16-tooth pulley, (b) GT2 60-tooth pulley, (c) GT2 300 mm timing belt, (d) 17HS4401 stepper motor, (e) JGA25-370 conveyor motor, (f) YYP370-12E vacuum pump, and (g) ZYE1-0530Z suction actuator

The 17HS4401 motor used for the robot axes is a 2-phase hybrid stepper motor with a step angle of 1.8° per step (200 steps per revolution), a rated voltage of 3 V, and a rated current of 1.5 A per phase. Based on these specifications, the TB6600 driver is used to provide adjustable output current from 0.5 A to 4 A, with microstepping selectable from full step to 1/16 step. The conveyor is driven by the JGA25-370 DC motor, while the YYP370-12E pump supplies the vacuum for the suction actuator.

The vision and operator-interface components provide the sensing and monitoring functions of the system. The DAHUA HTI-UC320 camera is used for image acquisition, the E3F-DS30C4 sensor detects workpiece presence, and the Weintek MT8071iP HMI provides an interface for monitoring and setting operating parameters. These components are shown together in Fig. 7.

Fig. 7. Vision, sensing, and HMI components: (a) DAHUA HTI-UC320 camera, (b) E3F-DS30C4 sensor, and (c) Weintek MT8071iP HMI

HMI display provides the operator with access to operating parameters and system status during system operation. Corresponding HMI display is shown in Fig. 8.

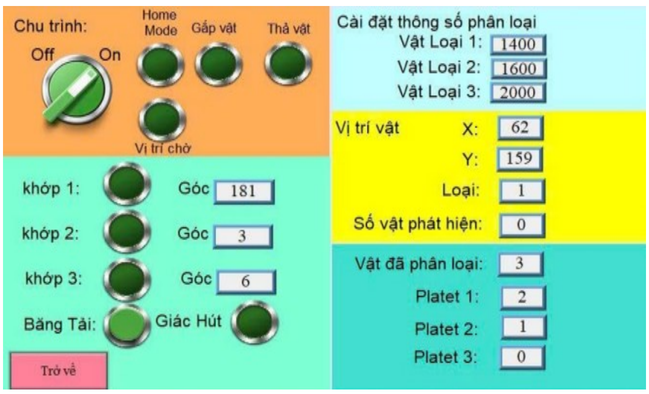

Fig. 8. HMI display for setting and monitoring operating parameters

III. METHOD

A. Vision-to-Robot Transformation and Image Processing

The first method converts camera observations into robot coordinates and object descriptors for sorting. The procedure comprises calibration, planar coordinate transformation, image processing, contour measurement, and classification, following the implementation documented in the thesis.

1. Camera Calibration

Prior to object detection, camera calibration is performed using a checkerboard pattern and OpenCV's cv2.calibrateCamera() function to determine the intrinsic matrix K and distortion coefficients (distCoeffs) isn shown in (14):

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

where f_x, f_y represent the focal lengths in pixels, and (c_x, c_y) is the principal point. The estimated intrinsic matrix and distortion coefficients are then applied to correct image distortion before converting 2D image coordinates to 3D real-world space.

2. Camera-To-Robot Coordinate Transformation

Since all workpieces lie on a fixed working plane, a planar homography transformation maps image pixel in (15) coordinates (u, v) to the robot coordinate space (X_r, Y_r):

$$s \begin{bmatrix} X_r \\ Y_r \\ 1 \end{bmatrix} = H_{\text{cam}\to\text{robot}} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{15}$$

Here, $H_{\text{cam}\to\text{robot}}$ is a 3×3 matrix estimated using nine reference points on the work table, exceeding the theoretical minimum of four points to enhance accuracy. This transformation accounts for both rotation and translation between the camera calibration pattern and the robot base frame.

3. Image Preprocessing and Otsu Thresholding

The vision pipeline applies background differencing, grayscale conversion, and Gaussian filtering before segmenting the image with Otsu's thresholding (sensitivity threshold set to 22). Contours are then extracted from the resulting binary image and filtered based on surface area, aspect ratio, and boundary constraints.

4. Contour Area, Centroid, and Classification

For each valid contour, the pixel centroid (c_x, c_y) is derived from spatial moments in (16):

$$c_x = \frac{M_{10}}{M_{00}}, \quad c_y = \frac{M_{01}}{M_{00}} \tag{16}$$

$$c_x = M_{10} / M_{00}, \quad c_Y = M_{01} / M_{00} \tag{17}$$

Contour area, computed via cv2.contourArea() in (18) is scaled to physical dimensions using a conversion factor k_{area}:

$$A_{\text{real}} = A_{\text{pixel}} \times k_{\text{area}} \tag{18}$$

The centroid is transformed to robot coordinates and the measured area is used to classify small, medium, or large objects before PLC sorting. The complete vision-to-robot processing sequence and its implementation flow are summarized in Fig. 9 and Fig. 10, respectively.

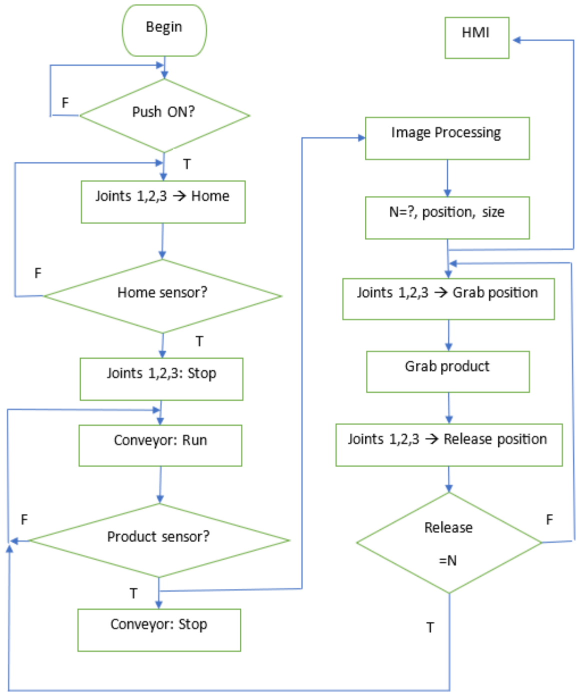

Fig. 9. System algorithm flowchart

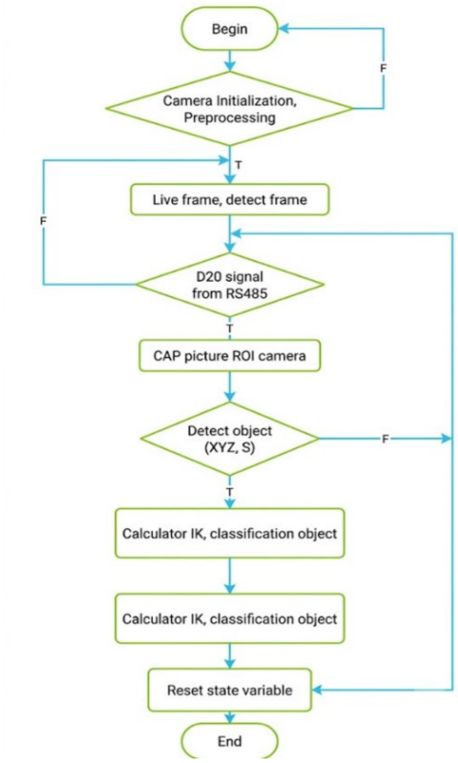

Fig. 10. Image processing algorithm flowchart

B. Jacobian-Based Trajectory Formulation

The second method extends the same D-H model to velocity-level Cartesian trajectory generation. For a desired Cartesian position trajectory $r_d(t) = [x_d, y_d, z_d]^T$, the translational velocity satisfies the formula (19):

$$\dot{r}_d = J_v(\theta)\dot{\theta} \tag{19}$$

Using the pseudoinverse, the corresponding joint velocity is shown in (20):

$$\dot{\theta} = J_v^+(\theta)\dot{r}_d \tag{20}$$

The resulting joint trajectory can be integrated subject to joint limits and Jacobian conditioning; for this 3-DOF robot, the formulation targets Cartesian position rather than full 6D pose.

The geometric Jacobian of the D-H model is shown in (21):

$$\begin{cases} J(\theta) = [J_v; J_\omega] \\ J_v = [z_0 \times (p_3 - p_0) \\ z_1 \times (p_3 - p_1) \\ z_2 \times (p_3 - p_2)] \\ J_\omega = [z_0, z_1, z_2] \end{cases} \tag{21}$$

The Jacobian provides a basis for trajectory tracking and velocity-level control, but the present experiment uses coordinate transformation, inverse kinematics, and PLC commands; no Jacobian-based tracking result is claimed.

IV. Experiment

The prototype was evaluated under laboratory conditions using the fabricated robotic sorting model and the measurement procedures described below. The fabricated prototype used for the experimental evaluation is shown in Fig. 11.

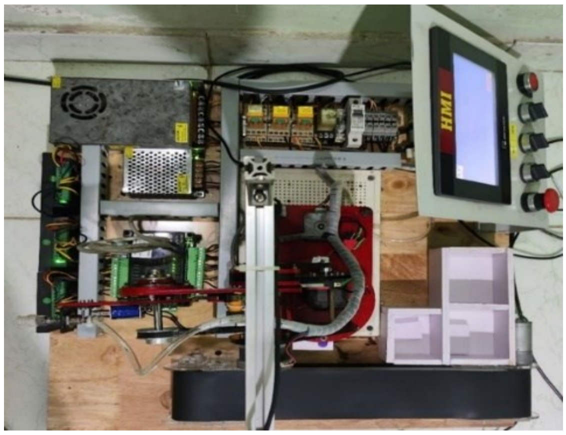

Fig. 11. Fabricated prototype

The experimental evaluation covers coordinate localization, object-area measurement, and size-based classification. The corresponding results are summarized in Table 3 and Table 4.

Table 3. Image Location

Point	Real Coordinates (mm)		Camera Coordinates (mm)		Error (mm)	
	X	Y	X	Y	X	Y
1	150	40	157	35	7	5
2	150	80	154	77	4	-3
3	150	120	155	110	5	-10
4	200	40	210	42	10	2
5	200	80	207	85	7	5
6	200	120	212	124	12	4
7	200	40	216	45	16	5
8	200	80	219	88	19	8
9	200	120	218	134	18	14
10	200	150	225	160	25	10
			Ratio		1.31	1.28

Table 3 compares the reference coordinates with the camera-derived coordinates and reports the corresponding localization errors. Point 10 shows the largest X-coordinate deviation, 25 mm, whereas the corresponding Y-coordinate deviation is 10 mm. Across the ten points, the mean absolute X- and Y-coordinate deviations are 12.3 mm and 6.6 mm, respectively. These localization errors should not be conflated with the reported ±2 mm average size-measurement error. The present experiment does not isolate lens distortion, illumination, contour localization, and calibration residuals separately; therefore, no single cause is assigned to the Point 10 outlier.

The comparison quantifies the deviation between the reference coordinates and the coordinates estimated from the camera-based system. The measured object areas at four rotation angles for the three tested object sizes are reported in Table 4.

Table 4. Experimental Classification Bounds

Product	Nominal Area (mm²)	Classification Range (mm²)	Error	Type
Product A	30x45 1350	1350-1450	10%	Small
Product B	35x45 1575	1475-1575	10%	Medium
Product C	40x50 2000	2000-2100	10%	Large

The classification ranges used for the three product groups are summarized in Table 5. These intervals are treated as the experimental decision ranges; the available material does not provide a derivation that would justify interpreting the listed limits as formal ±10% tolerance bounds.

Table 5. Measured Object Areas at Various Rotation Angles

Object	Shape	Rotation (°)	Area (mm²)
Object 1	Rectangle	0	1355
	30 × 45 × 10 mm	45	1362
	Area 1350 mm²	60	1401
		90	1360
Object 2	Rectangle	0	1601
	35 × 45 × 10 mm	45	1573
	Area 1575 mm²	60	1653
		90	1581
Object 3	Rectangle	0	2004
	40 × 50 × 10 mm	45	2052
	Area 2000 mm²	60	2043
		90	2009

The measured areas remain close to the nominal areas across the tested rotation angles, providing the basis for the size-based classification used in the experiment.

The reported experimental results indicate an average size-measurement error of approximately ±2 mm, a classification accuracy of about 95%, and an average processing time of 2-3 s per product. The available thesis material does not specify the number of classification trials underlying the approximately 95% figure; therefore, this value is presented as a reported experimental accuracy rather than as a statistically estimated performance measure. The observed residual errors may be influenced by calibration residuals, illumination, and image-localization effects; however, these factors were not isolated experimentally.

Under the laboratory conditions evaluated, the integrated vision, PLC, HMI, and robotic subsystems provide a practical workflow for automated size-based sorting. The combination of image processing and robotic manipulation demonstrates the feasibility of the proposed laboratory-scale system, while broader industrial deployment requires additional validation under varied operating conditions.

1. *Comparison with Recent Vision-Guided Robotic Systems*

Recent studies demonstrate a shift toward learning-based perception and explicit robot-camera calibration. Li *et al.* [16] presented a learning-based 3D hand-eye calibration method for estimating the camera-to-robot transformation. Dirr *et al.* [17] demonstrated CNN-based instance segmentation integrated with robotic picking for industrial parts, while Deng *et al.* [18] reported a dual-stream YOLO-based grasping system for box-shaped objects. In contrast, the present study uses a three-DOF arm and deterministic OpenCV processing with checkerboard calibration. The comparison is qualitative because the referenced studies use different robots, sensors, datasets, object classes, and evaluation protocols.

2. *Dynamic-Testing Limitation*

The reported 2-3 s processing time was obtained under laboratory conditions. Systematic experiments at multiple conveyor speeds were not included in the available dataset; therefore, dynamic pick accuracy, speed-dependent processing latency, and throughput under moving-conveyor conditions are not claimed. Future experiments should report conveyor speed, image-acquisition time, processing time, communication delay, robot motion time, and successful pick rate.

V. Conclusion

In this study, an image-processing-based robotic arm classification system was designed and implemented by integrating camera-based image acquisition, OpenCV-based object detection and size measurement, coordinate transformation, robotic control, and an HMI interface for monitoring. Under the reported laboratory conditions, the system achieved a classification accuracy of approximately 95% and demonstrated size-based sorting using the integrated vision and robotic subsystems. These results support the feasibility of the proposed laboratory-scale system for automated sorting applications. However, the present implementation remains limited to size-based classification and fixed camera conditions, and its performance under broader industrial conditions has not yet been established. Future work will focus on improving robustness through multi-feature recognition, including color, shape, and defect detection, incorporating 3D vision for improved spatial measurement, optimizing image processing for real-time operation, and developing more flexible interfaces for expanded applications.

Acknowledgement

We want to give thanks to Ph.D. Van-Dong-Hai Nguyen (HCM-UTE) for his supervision. Link to the operation is: https://www.youtube.com/watch?v=IjtEABiRqMg&feature=youtu.be.

REFERENCES

[1] L. Fernandes and B. R. Shivakumar, “Identification and Sorting of Objects based on Shape and Colour using robotic arm,” in *Proceedings of the 4th International Conference on Inventive Systems and Control, ICISC 2020*, pp. 866–871, 2020, https://doi.org/10.1109/ICISC47916.2020.9171196.

[2] V. Pereira, V. A. Fernandes, and J. Sequeira, “Low cost object sorting robotic arm using Raspberry Pi,” in *2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite, GHTC-SAS 2014*, pp. 1–6, 2014, https://doi.org/10.1109/GHTC-SAS.2014.6967550.

[3] O. K. Meng, O. Pauline, L. E. Soong, and S. C. Kiong, “Robotic arm system with computer vision for colour object sorting,” *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 50–56, 2018, https://doi.org/10.14419/ijet.v7i4.27.22479.

[4] A. S. Shaikat, S. Akter, and U. Salma, “Computer Vision Based Industrial Robotic Arm for Sorting Objects by Color and Height,” *Journal of Engineering Advancements*, vol. 01, no. 04, pp. 116–122, 2020, https://doi.org/10.38032/jea.2020.04.002.

[5] X. Ye, Y. Zhou, H. Guo, and Z. Luo, “A computer vision-based approach to automatically extracting the aligning information of precast structural components,” *Automation in Construction*, vol. 164, p. 105478, 2024, https://doi.org/10.1016/j.autcon.2024.105478.

[6] H. Q. T. Ngo, “Using HSV-based approach for detecting and grasping an object by the industrial mechatronic system,” *Results in Engineering*, vol. 23, p. 102298, 2024, https://doi.org/10.1016/j.rineng.2024.102298.

[7] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, and R. Khan, “Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 180–186, 2022, https://doi.org/10.18196/jrc.v3i2.13906.

[8] O. Aburas, H. M., A. Elnageh, M. Ebshish, and M. Algadi, “Designing and Implementing a 3DOF Robotic Arm for Color Sorting and Object Identification Using Computer Vision Technology,” *The International Journal of Engineering & Information Technology (IJEIT)*, vol. 12, no. 1, pp. 71–78, 2024, https://doi.org/10.36602/ijeit.v12i1.479.

[9] C. Kaymak and A. Ucar, “Implementation of Object Detection and Recognition Algorithms on a Robotic Arm Platform Using Raspberry Pi,” in *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, Malatya, Turkey, 2019, pp. 1–8, https://doi.org/10.1109/IDAP.2018.8620916.

[10] J. Wu, “Enhancing Object Sorting Under Low-Light Conditions with CLAHE, Gaussian Blur, ROI, and Custom PID on a Raspberry Pi Robotic Arm,” *Applied and Computational Engineering*, vol. 96, no. 1, pp. 93–98, 2024, https://doi.org/10.54254/2755-2721/96/20241240.

[11] J. M. S. T. Motta, G. C. De Carvalho, and R. S. McMaster, “Robot calibration using a 3D vision-based measurement system with a single camera,” *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 6, pp. 487–497, 2001, https://doi.org/10.1016/S0736-5845(01)00024-2.

[12] S. W. Wijesoma, D. F. H. Wolfe, and R. J. Richards, “Eye-to-Hand Coordination for Vision-Guided Robot Control Applications,” *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 65–78, 1993, https://doi.org/10.1177/027836499301200105.

[13] D. Wang, Y. Bai, and J. Zhao, “Robot manipulator calibration using neural network and a camera-based measurement system,” *Transactions of the Institute of Measurement and Control*, vol. 34, no. 1, pp. 105–121, 2012, https://doi.org/10.1177/0142331210377350.

[14] M. Bowman and A. Forrest, “Transformation calibration of a camera mounted on a robot,” *Image and Vision Computing*, vol. 5, no. 4, pp. 261–266, 1987, https://doi.org/10.1016/0262-8856(87)90002-3.

[15] A. K. Singh and M. S. R. Ali, “Automatic sorting of object by their colour and dimension with speed or process control of induction motor,” in *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2017*, 2017, pp. 1–6, https://doi.org/10.1109/ICCPCT.2017.8074289.

[16] L. Li, X. Yang, R. Wang, and X. Zhang, “Automatic Robot Hand-Eye Calibration Enabled by Learning-Based 3D Vision,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 110, no. 3, p. 130, 2024, https://doi.org/10.1007/s10846-024-02166-4.

[17] J. Dirr, J. C. Bauer, D. Gebauer, and R. Daub, “Cut-paste image generation for instance segmentation for robotic picking of industrial parts,” *International Journal of Advanced Manufacturing Technology*, vol. 130, no. 1–2, pp. 191–201, 2024, https://doi.org/10.1007/s00170-023-12622-4.

[18] H. Deng, J. Shi, J. Cui, S. Bai, and X. Zuo, “A robotic grasping method of box-shaped objects based on Dual-Stream You Only Look Once framework,” *Engineering Applications of Artificial Intelligence*, vol. 159, p. 111559, 2025, https://doi.org/10.1016/j.engappai.2025.111559.