IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 3, June 2026, pp. 2325~2336
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i3.pp2325-2336

Yarn inspection and sorting system using robotic vision and machine learning

Emmanuel Agung Nugroho[1], Joga Dharma Setiawan[2], Deni Kurnia[1], Nanang Roni Wibowo[1]
[1]Department of Mechatronics Engineering, Politeknik Enjinering Indorama, Purwakarta, Indonesia
[2]Department of Mechanical Engineering, Faculty of Engineering, Universitas Diponegoro, Semarang, Indonesia

Article Info

Article history:

Received Jul 18, 2025
Revised Jan 23, 2026
Accepted Apr 20, 2026

Keywords:

4-DOF manipulator
Machine learning
Raspberry Pi
Robotic vision
Yarn inspection

ABSTRACT

The increasing demand for automation in the textile industry, particularly in quality inspection processes, underscores the need for intelligent and cost-effective solutions. Conventional methods of yarn classification and sorting remain labor-intensive, time-consuming, and susceptible to human error, resulting in inconsistent quality control. This study introduces an automated system for yarn inspection and sorting that integrates robotic vision, machine learning, and position-based visual servoing (PBVS) for real-time motion control. The proposed system combines Raspberry Pi-based machine learning with computer vision utilizing a 4-degree-of-freedom (4-DOF) robotic manipulator and a webcam, enabling precise pick-and-place operations based on yarn classification into four categories: good, striped, moldy, and dirty. Experimental results validate the system's effectiveness, achieving an average deviation of 0.375 mm along the x-axis, 0.69 mm along the y-axis, and 0.675 mm along the z-axis, resulting in an overall position error of 0.58 mm. These results demonstrate the system's robustness and reliability in dynamic industrial environments. The novelty of this research lies in leveraging a low-cost embedded architecture with advanced visual servoing for textile automation, reducing operational errors, improving efficiency, and supporting industry 4.0 adoption.

Corresponding Author:

Emmanuel Agung Nugroho
Department of Mechatronics Engineering, Politeknik Enjinering Indorama
Purwakarta 41152, West Java, Indonesia
Email: emmanuel.agung@pei.ac.id

1. INTRODUCTION

The textile industry is undergoing a rapid transformation driven by global trend toward industry 4.0, where automation and intelligent systems are reshaping conventional manufacturing processes [1]. Among the critical areas requiring improvement is the inspection and sorting of yarns, which plays a vital role in determining product quality and production efficiency. Traditionally, yarn inspection and sorting are performed manually by human operators based on visual assessment. However, this approach is often prone to human error, inconsistent judgment, high labor costs, and significant time consumption, especially when dealing with high production volumes and complex defect variations [2], [3].

Recent advances in robotics and computer vision technologies have enabled the development of compact and effective solutions to automate these repetitive tasks. In particular, the integration of robotic manipulators with vision systems powered by machine learning has emerged as a promising approach to improve precision, consistency, and decision-making capabilities in material handling applications [4], [5]. These systems are often supported by low-cost platforms such as the Raspberry Pi, which provides sufficient computing power for real-time processing and control, making them suitable for automation in small-to medium-scale industrial environments [6].

One of the key techniques bridging visual perception with robotic control in such systems is visual servoing. Visual servoing refers to the use of camera-based visual feedback to control the motion of robotic manipulators in real-time. This technique is generally classified into two main types: image-based visual servoing (IBVS), in which control laws are derived directly from image coordinates, and position-based visual servoing (PBVS), which utilizes 3D position estimation of objects to guide robotic movements [7]. In the context of yarn sorting, visual servoing allows dynamic adjustment of the end-effector's position and orientation based on continuously updated visual input, leading to higher accuracy and better adaptability compared to conventional open-loop control systems [8].

This study focuses on the development and evaluation of a PBVS system for yarn sorting, which integrates a 4-degree-of-freedom (4-DOF) robotic manipulator controlled by a Raspberry Pi with a vision-based classification module. The system is designed to detect various yarn categories including good, striped, moldy, and dirty using machine learning algorithms trained on a dataset of yarn images. Based on the classification output, the robotic manipulator performs a pick-and-place operation to sort each yarn sample into its corresponding storage bin [9], [10]. The novelty of this research lies in the integration of a Raspberry Pi platform to coordinate a 4-DOF robotic manipulator (Dobot Magician) for autonomous operation [8], [11]. Implementation of a webcam-based image acquisition system supported by machine learning for real-time yarn quality classification [12]. The application of visual servoing to enable dynamic manipulation based on visual feedback, allowing the robot to retrieve objects from a fixed conveyor endpoint and place them into four destination bins [13]. System performance is measured through positional deviation analysis to validate precision and consistency, targeting an average placement error of less than 1 mm [14].

2. METHOD

Visual servoing is closed-loop control method that uses camera-based visual feedback to guide robotic actuators with precision, integrating machine learning, image processing, and robotic kinematics [15], [16]. There are two primary implementations of visual servoing: PBVS and IBVS. This study adopts the PBVS method, as illustrated in Figure 1.

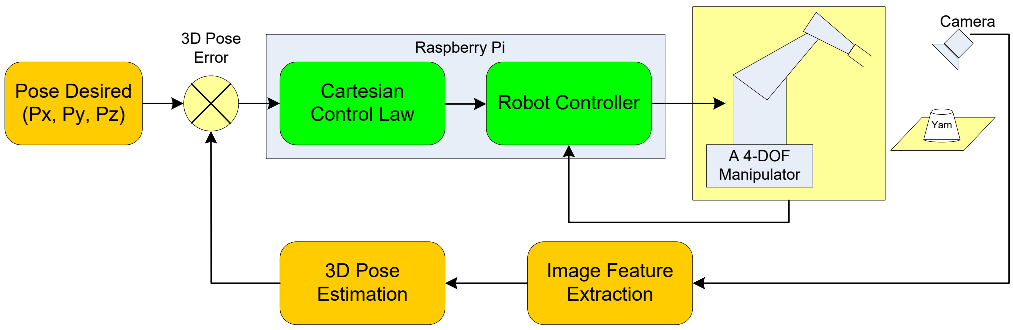


Figure 1. Schematic of the PBVS approach

Figure 1 shows the PBVS control strategy, where robot motion is guided by a predefined target pose for object placement. The system estimates the object's 3D pose comprising position and orientation from camera-based visual classification and generates control signals to minimize the error between actual and reference poses. The architecture consists of three main components:

i) Camera: captures real-time images of the yarn and acquires sample data for visual processing [17].
ii) Raspberry Pi: serves as the central control unit responsible for image processing, classification, and issuing actuation commands [18].
iii) Dobot Magician: 4-DOF robotic manipulator used to perform pick-and-place operations based on the classification results [19].

Image classification uses OpenCV and a TensorFlow model trained via Google Teachable Machine to categorize yarn into good, striped, moldy, and dirty. The PBVS loop computes transformation parameters after classification and commands the manipulator to sort each yarn into its designated bin [20].

2.1. Workflow and implementation process

In this study, labeled yarn datasets were stored in Raspberry Pi for model training. Upon capturing a new image, the system classifies the yarn by comparing it with the trained model and selects the class with the highest confidence. After classification, Raspberry Pi sends commands to the 4-DOF manipulator for sorting. Figure 2 illustrates the PBVS-based image processing and sorting mechanism controlled by Raspberry Pi.

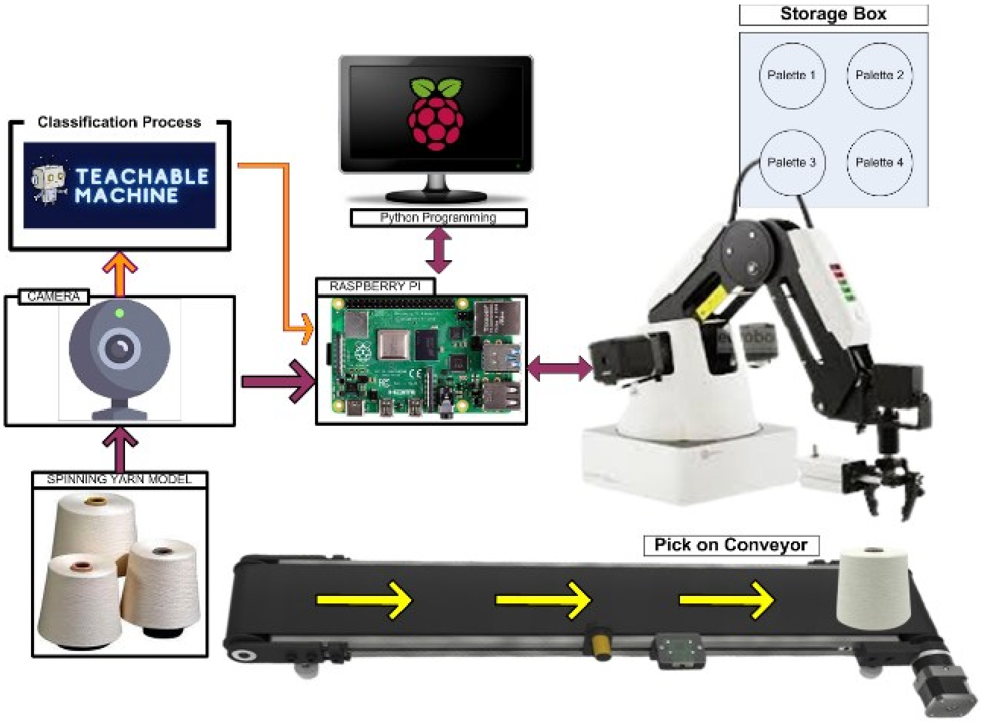


Figure 2. Mechanism of the yarn sorting system using PBVS method

Figure 2 illustrates the implementation of the visual servoing system controlled by Raspberry Pi, which manages visual input, classification, and manipulator actuation. The hardware setup includes a camera for image capture, Raspberry Pi for processing and control, a 4-DOF Dobot Magician for pick-and-place tasks, and a conveyor for yarn transportation to the inspection area. The implementation of the proposed yarn sorting system follows a structured workflow consisting of five main stages, as detailed as follows.

i) Image acquisition: the process starts by capturing real-time yarn images on the conveyor with a fixed camera, providing essential visual data on position, shape, and texture [21].
ii) Image processing and classification [22]: processing images via OpenCV and a TensorFlow Lite model trained with Google Teachable Machine to classify yarn as good, striped, moldy, or dirty, with results indicated through light emitting diodes (LEDs) on Raspberry Pi general purpose input/output (GPIO) [23].
iii) Position and orientation calculation: based on the classification output and pixel coordinates from the camera frame, the system computes the desired position and orientation of manipulator's end-effector in world coordinates [24].
iv) Manipulator control: Raspberry Pi generates commands for the 4-DOF Dobot Magician to perform pick-and-place operations [25], [26].
v) Visual feedback and verification: a final image is captured after the actuation to verify the success of the pick-and-place operation. This feedback loop ensures that the manipulator reaches the correct target location and that the yarn has been sorted into the correct category [27].

The system workflow of a 4-DOF manipulator control system using Raspberry Pi is shown in Figure 3. This figure outlines the visual data flow, decision logic, and manipulator motion sequence from image acquisition to final object placement. The entire process is designed for high precision and real-time execution, enabling efficient and accurate yarn classification in industrial environment.

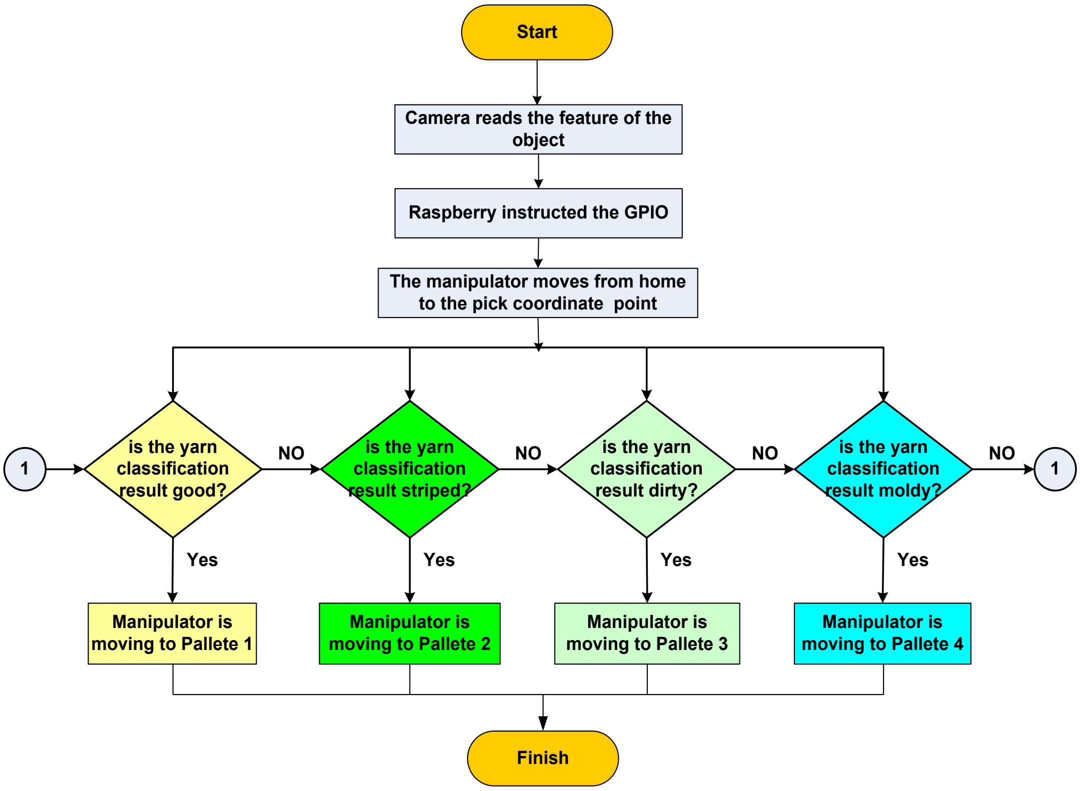


Figure 3. Flowchart of a 4-DOF manipulator control system using Raspberry Pi

2.2. Hardware setup and interface design

This section explains the communication interface between Raspberry Pi 4B and the Dobot Magician robotic arm, focusing on programming architecture, GPIO mapping, and supporting libraries. Raspberry Pi serves as the central controller, handling image capture, classification, and actuation. Python was selected for its compatibility with OpenCV (image processing), TensorFlow (on-device inference), and PySerial (USB-based serial communication for motion commands). Table 1 shows the GPIO configuration on Raspberry Pi, where each pin is assigned for actuator control (LEDs, buzzer) or input triggers.

Table1. GPIO addressing for yarn inspection system

No	GPIO number	Function	Description
1	17	Good yarn	Activated when a good-quality yarn is detected
2	18	Striped yarn	Activated when striped yarn is detected
3	27	Moldy yarn	Activated when moldy yarn is detected
4	22	Dirty yarn	Activated when dirty yarn is detected
5	23	Green LED	Remains on when the production output is good
6	24	Red LED	Turns on when a defect is detected
7	25	Buzzer	Activates emergency audio alert
8	26	Push button start	Initiates the classification and sorting process
9	19	Push button stop	Terminates the current process cycle

This GPIO configuration allowed the system to provide immediate visual and audio feedback during the yarn inspection process. The green LED signaled successful classification and sorting of acceptable yarn, while the red LED and buzzer were used to alert the presence of defective yarn. Furthermore, the push buttons served as user-controlled triggers to start or stop the automated process.

2.3. Dobot Magician control system

This section explains the integration of Raspberry Pi 4B with the 4-DOF Dobot Magician robotic arm using Python and the pydobot library, demonstrating low-cost microcontroller feasibility for intelligent robotic applications involving real-time image classification and actuation. Raspberry Pi functions as both the image inference processor and command center for robotic control. A key element is the camera_thread_function, which handles continuous image capture, classification inference, and corresponding robotic movements [28]. Real-time visual data is acquired via an OpenCV-based camera module (cv2.VideoCapture), at 640×480 resolution, balancing image quality and processing speed for efficient real-time operation [29], [30].

Table 2 summarizes the initialization of camera_thread_function, which operates in a separate thread to enable continuous image capture while executing robot control tasks. The use of stop_event allows safe termination during shutdown, while lock ensures thread-safe access to shared resources such as the camera stream and control signals. This threaded design enhances system responsiveness and stability during autonomous operation.

To respond effectively to classification results, the Dobot Magician robotic manipulator is programmed to execute precise movements using suction cup end effector. Two main functions, move_dobot and run_movements, convert classification labels into robot commands. The move_dobot function handles position-based motion and suction control by accepting (x, y, z) coordinates and a Boolean flag for activating or deactivating the suction. It also prints movement status to the terminal for real-time monitoring.

Table 2. Camera initialization and image capture functionality

Instruction	Description
def camera_thread_function (dobot, stop_event, lock)	Defines a function to handle image capture and robot control. It takes three parameters: dobot (robot object), stop_event (thread stop flag), and lock (synchronization mechanism).
lobal cap	Declares cap as a global variable to allow access outside the function.
cap = cv2.VideoCapture(0)	Initializes video capture from the default camera device (index 0).
cap.set (cv2.CAP_PROP_FRAME_WIDTH, 640)	Sets the video frame width to 640 pixels.
cap.set (cv2.CAP_PROP_FRAME_HEIGHT, 480)	Sets the video frame height to 480 pixels.

Table 3 summarizes the control instructions for the Dobot Magician’s movement and suction cup operation. The move_dobot function handles position-based motion and suction activation, providing real-time feedback via status messages. It moves the manipulator to specified (x, y, z) coordinates, activates or deactivates suction as required, and introduces a short delay to ensure task stability. After classification, the run_movements function maps each predicted label (good, stripe, moldy, dirty) to a predefined motion sequence for accurate sorting.

Table 3. Dobot movement and suction cup control instructions

Instruction	Description
def move_dobot (dobot, x, y, z, suction)	Defines a function to move the robot to specified coordinates and control the suction cup.
Print (f"Moving to: ({x}, {y}, {z}) with suction {'ON' if suction else 'OFF'}")	Displays a status message indicating the movement and suction status.
dobot.move_to(x, y, z, 0)	Moves the robot to the desired position. The final parameter (0) is the joint orientation.
if suction	Checks whether suction is needed. If True, activates suction; otherwise, deactivates it.
dobot.suck (True)	Activates the suction cup to grasp the object.
dobot.suck (False)	Deactivates the suction cup to release the object.
time.sleep (1)	Pauses the program for one second to ensure stable execution.

Table 4 presents the run_movements function, which maps each classification label “GOOD, STRIPE, MOLDY, DIRTY” to a specific motion routine using the move_dobot function. Each category triggers a predefined trajectory and suction control strategy, ensuring accurate placement or rejection. This implementation demonstrates dynamic motion generation based on visual classification, reducing manual inspection and enabling real-time defect sorting in industrial environments.

Table 4. Dobot motion execution based on label classification

Instruction	Description
def run_movements (dobot, label)	Defines a function that maps classification labels to movement routines.
if label == '0 GOOD'	Executes motion specific to 'GOOD' label.
elif label == '1 STRIPE'	Executes motion specific to 'STRIPE' label.
elif label == '2 MOLDY'	Executes motion specific to 'MOLDY' label.
elif label == '3 DIRTY'	Executes motion specific to 'DIRTY' label.
move_dobot (dobot, x, y, z, suction)	Calls the movement function with defined parameters for each label.

3. RESULTS AND DISCUSSION

This section evaluates the performance of the Dobot Magician in implementing vision-based motion control using Raspberry Pi as the main processing unit. The system leverages real-time visual feedback to obtain positional data, enabling precise and adaptive manipulator adjustments. The analysis focuses on the Dobot Magician's kinematics and motion execution performance, emphasizing positional accuracy and trajectory planning.

3.1. Kinematics analysis of Dobot Magician

3.1.1. Denavit-Hartenberg analysis of the 4-degree-of-freedom manipulator

To establish an accurate kinematic model of the Dobot Magician manipulator, a Denavit-Hartenberg (D-H) analysis was conducted. The robot comprises five major components: static base, rotating base, shoulder, elbow, and wrist. Each of these components was modeled in autodesk inventor based on the original physical dimensions and then assembled in accordance with the coordinate frame structure provided by the Dobot manufacturer [31]. The resulting assembly, including the joint reference frames, is shown in Figure 4.

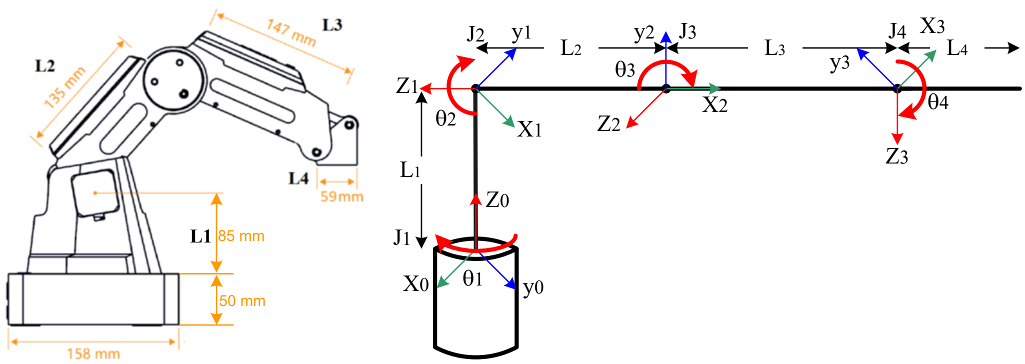


Figure 4. Structure and coordinat frame of Dobot Magician

Based on this structural layout, the D-H parameter modeling for the 4-DOF configuration excludes the fifth frame corresponding to the end-effector. The parameter set summarized in Table 5 defines the relative transformation between each joint using the standard four D-H parameters: link length L_i, link twist α_i, link offset d_i and joint angle θ_i. These parameters were defined for each joint of the 4-DOF configuration of the Dobot Magician, excluding the end-effector at the wrist.

Table 5. D-H parameters of the 4-DOF Dobot Magician manipulator

Frame i	L_i	α_i	d_i	θ_i
1	0	90°	85	$\theta_1 = 135^o to\, 135^o$
2	135	0°	0	$\theta_2 = 0^o to\, 85^o$
3	147	0°	0	$\theta_2 = 0^o to\, 85^o$
4	59	0°	0	$\theta_4 = 90^o to\, -90^o$
5	-	90°	-70	$\theta_5 = 0^o$

The D-H method derives a homogeneous matrix formulation relative to each manipulator frame. Thus, for a 4-DOF manipulator such as Dobot Magician, it yields as in (1).

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i\, C\alpha_i & S\theta_i\, S\alpha_i & L_i C\theta_i \\ S\theta_i & C\theta_i\, C\alpha_i & -C\theta_i\, S\alpha_i & L_i s\theta_i \\ 0 & S\alpha_i & C\theta_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

From the homogeneous equation, the matrix connecting frame 1 to frame 5 is obtained as (2).

$$A_5^0 = A_1^0 . A_2^1 . A_3^2 . A_4^3 . A_5^4 \tag{2}$$

In this provision, the kinematics of Dobot Magician is formed by setting the link lengths L_1 =85 mm, L_2 =135 mm, L_3 =147 mm, and L_4 =59 mm, with joint variables θ_i derived from the manipulator's actuation based on its slider input. Each transformation matrix A_i^{i-1} is constructed based on the respective D-H parameters. In this case the configuration of joint angles: θ_1=-33°, θ_2=-7°, θ_3=-24°, and θ_4=-43°.

So, the transformation matrices from base to end-effector are computed step-by-step. The homogeneous transformation matrices derived are as follows:

θ_1=-33°, Cos θ_1=0.838, Sin θ_1=-0.544

$$A_1^0 = \begin{bmatrix} 0.838 & 0 & -0.544 & 73.61 \\ S\theta_i & 0 & 0.838 & 42.5 \\ 0 & -1 & 0 & 85 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_2=-37°, Cos θ_2=0.992, Sin θ_2=-0.122

$$A_2^1 = \begin{bmatrix} 0.992 & 0 & 0.122 & 133.99 \\ -0.122 & 0 & 0.992 & -16.47 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_3=-24°, Cos θ_3=0.913, Sin θ_3=-0.406

$$A_3^2 = \begin{bmatrix} 0.913 & 0 & 0.406 & 134.21 \\ -0.406 & 0 & 0.913 & -59.68 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_4=-43°, Cos θ_4=0.731, Sin θ_4=-0.681

$$A_4^3 = \begin{bmatrix} 0.731 & 0 & 0.681 & 43.13 \\ -0.681 & 0 & 0.731 & -40.18 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -70 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall transformation from base to wrist is obtained by:

$$A_5^0 = \begin{bmatrix} 0.231 & -0.544 & -0.806 & 269.82 \\ -0.150 & -0.838 & 0.523 & -91.76 \\ -0.961 & 0 & -0.275 & -63.88 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.1.2. Forward kinematics analysis of the 4-degree-of-freedom manipulator

In the context of a 4-DOF manipulator, forward kinematics is essential for determining the position and orientation of the end-effector based on the joint angles of the manipulator [32]. This process involves a closed-loop vector analysis to derive the position equations. The number of equations depends on the number of joints and the length of each manipulator link.

The forward kinematics equation is defined as in (3).

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} C(\theta_1).\left(L_2C(\theta_2) + L_3C\,(\theta_2+\theta_3) + L_4C\,(\theta_2+\theta_3+\theta_4)\right) \\ S(\theta_1).\left(L_2C(\theta_2) + L_3C\,(\theta_2+\theta_3) + L_4C\,(\theta_2+\theta_3+\theta_4)\right) \\ L_1 + \left(L_2S(\theta_2) + L_3S\,(\theta_2+\theta_3) + L_4S\,(\theta_2+\theta_3+\theta_4)\right) \end{bmatrix} \tag{3}$$

The required trigonometry values are calculated:

$$Cos(\theta_2+\theta_3) = 0.857$$
$$Cos(\theta_2+\theta_3+\theta_4) = 0.276$$
$$Sin(\theta_2+\theta_3) = -0.5151$$
$$Sin(\theta_2+\theta_3+\theta_4) = -0.961$$

Now calculate each component of the position vector based on the forward kinematics equation:

$$P_x = 0.838 \times (135 \times 0{,}992 + 147 \times 0.857 + 59 \times 0.276)$$
$$P_x = 0.838 \times (133.92 + 125.88 + 16.28)$$
$$P_x = 0.838 \times 276.08 = 231.25\, mm$$

$$P_y = -0.544 \times (135 \times 0.992 + 147 \times 0{,}857 + 59 \times 0{,}276)$$
$$P_y = -0.544 \times 276.08 = -150.17\, mm$$

$$P_z = 85 + 135 \times (-0.122) + 147 \times (-0.515) + 59 \times (-0.961)$$
$$P_z = 85 - 16.47 - 75.71 - 56.70 = 63.88\, mm$$

So, the final position on forward kinematics is:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} 231.25 \\ -150.17 \\ -63.88 \end{bmatrix}$$

3.1.3. Inverse kinematics analysis of the 4-degree-of-freedom manipulator

The inverse kinematics analysis is used to determine the joint angles θ_1, θ_2, θ_3, and θ_4 of a 4-DOF manipulator [33]. This analysis uses a geometric and trigonometric approach. Given the desired position of the end-effector (P_x, P_y, and P_z):

i) Calculated of θ_1 as defined in (4).

$$tan\ \theta_1 = \frac{P_y}{P_x}$$
$$tan\ \theta_1 = \frac{-150.36}{231.69} \approx -0.649$$
$$\theta_1 = tan^{-1}(-0.649) \approx -33° \tag{4}$$

ii) Preparation for θ_2, and θ_3:

$$\alpha = \sqrt{P_x^2 + P_y^2} = \sqrt{(231.69)^2 + (-150.36)^2}$$
$$\alpha \approx 276.39\, mm$$
$$b = P_z - L_1 = -63.88 - 85 = 148.88\, mm$$

iii) Calculation of θ_3 as defined in (5).

$$cos(\theta_3) = \frac{a^2+b^2-L_2^2-L_3^2}{2L_2L_3}$$
$$cos(\theta_3) = \frac{(276.39)^2+(-148.88)^2-135^2-147^2}{2\times135\times147}$$
$$cos(\theta_3) = \frac{58733}{39690} \approx 0.741$$
$$sin(\theta_3) = \sqrt{1-(0.741)^2} \approx \sqrt{0.451} \approx 0.671$$
$$\theta_3 = tan^{-1}\left(\frac{0.671}{0.741}\right) \approx tan^{-1}(0.9069) \approx 42.6° \tag{5}$$

iv) Calculation of θ_2 as defined in (6).

$$\theta_2 = tan^{-1}\left(\frac{b}{a}\right) - tan^{-1}\left(\frac{L_3 sin(\theta_3)}{L_2+L_3cos(\theta_3)}\right)$$
$$\theta_2 = tan^{-1}\left(\frac{-148.88}{a276.39}\right) - tan^{-1}\left(\frac{147\times0.671}{135+147\times0.741}\right)$$
$$\theta_2 = tan^{-1}(-0.5388) - tan^{-1}(0.4)$$
$$\theta_2 \approx -28.2 - 21.8 = -50° \tag{6}$$

v) Calculation of θ_4: assuming the manipulator wrist maintains a flat orientation (zero end-effector pitch), the last joint compensates for the sum of the previous angles as in (7).

$$\theta_4 = (\theta_2 + \theta_3)$$
$$\theta_4 = (-50° - 42.6°) = -7.4° \quad (7)$$

The slight difference between the inverse and forward values particularly in θ_4 may result from simplifications in the inverse kinematics method, which may not fully account for the end-effector's orientation, or differences in reference frame conventions and angular decompositions.

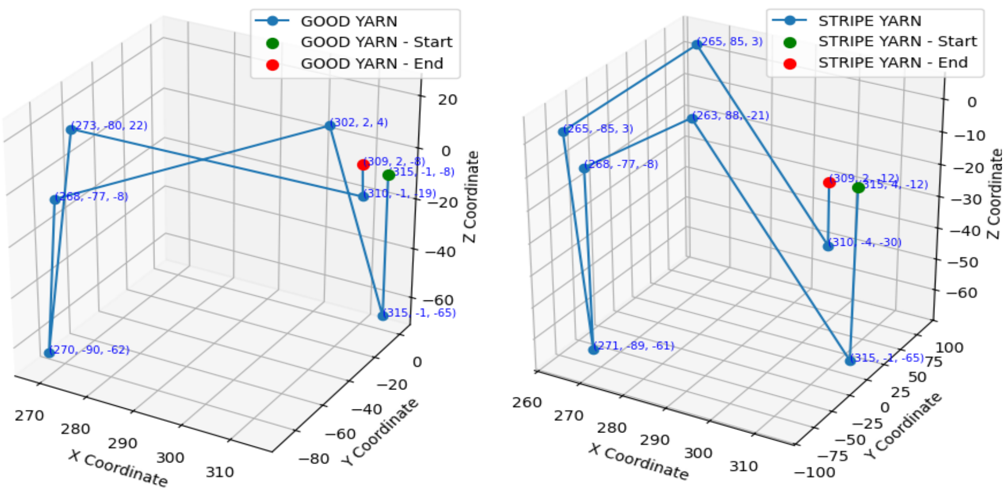
3.2. Performance analysis of Dobot Magician movement

The experiment evaluated pick-and-place tasks for yarn spools classified into four categories good, striped, moldy, and dirty, using a vision system powered by Teachable Machine. Objects were detected via a webcam, classified with machine learning, and positioned by the Dobot Magician at predefined target coordinates. The pickup point for all yarn types was fixed at (315, -1, and -65), while destination points varied by category. Table 6 compares programmed and actual end-effector coordinates.

Table 6. Comparison of intended vs. actual end-effector coordinates

Category (yarn)	Initial coordinate (mm)	Final coordinate (programmed) (mm)	Final coordinate (actual (mm))
Good	(315, -1, -65)	(270, -90, -62)	(269.9, -90.2, -61.8)
Striped	(315, -1, -65)	(270, 90, -62)	(270.1, 89.8, -61.9)
Moldy	(315, -1, -65)	(170, -90, -62)	(170.1, -89.9, -60.8)
Dirty	(315, -1, -65)	(170, 90, -62)	(168.8, 91.3, -60.8)

The results demonstrate high precision in the manipulator's performance. Results show minimal deviation between intended and actual positions, confirming high accuracy x-axis: 0.375 mm, y-axis: 0.69 mm, z-axis: 0.675 mm, and overall mean deviation: ±0.58 mm. In addition to endpoint accuracy, an evaluation of the manipulator's motion trajectory was conducted to observe movement patterns during sorting tasks. Figure 5 presents the trajectory behavior for two classification cases: Figure 5(a) shows the movement path when handling good yarn, while Figure 5(b) illustrates the trajectory generated during the sorting of striped yarn. Differences in placement coordinates along the x and y axes result in distinct paths for each category. These movements require coordinated multi-axis control involving object gripping, vertical lifting, horizontal translation, and controlled lowering toward the designated target location.



(a) (b)

Figure 5. Trajectory of the manipulator moving of (a) good yarn and (b) striped yarn

Trajectory analysis confirms that variations occur across categories due to x-y placement differences. The coordinated movements gripping, lifting, horizontal displacement, and precise lowering as shown in Figure 5, validate the system's capability to perform dynamic, real-time operations with visual classification. These results demonstrate precision, responsiveness, and adaptability for industrial automation environments.

4. CONCLUSION

The developed yarn inspection and sorting system successfully integrates robotic vision, machine learning, and PBVS to achieve real-time and precise operation. Experimental results indicate high positional accuracy, with an overall mean deviation of ±0.58 mm (x-axis: 0.375 mm, y-axis: 0.69 mm, z-axis: 0.675 mm), demonstrating reliable pick-and-place performance using a 4-DOF manipulator. The system achieved 100% successful sorting across four yarn categories (good, striped, moldy, and dirty) and executed classification and actuation cycles in under 2 seconds per operation, validating its efficiency for dynamic industrial environments. These outcomes confirm the feasibility of using low-cost embedded platforms for intelligent quality control and material handling in industry 4.0 applications.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Emmanuel Agung Nugroho	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Joga Dharma Setiawan		✓				✓		✓		✓		✓		✓
Deni Kurnia	✓		✓	✓		✓	✓			✓	✓	✓	✓	
Nanang Roni Wibowo	✓		✓	✓	✓	✓	✓		✓		✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board or equivalent committee.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [EAN], upon reasonable request.

REFERENCES

[1] Ş. Çiğdem, I. M. Kavaliauskiene, and B. Yıldız, "Industry 4.0 and industrial robots: a study from the perspective of manufacturing company employees," *Logistics*, vol. 7, no. 1, Mar. 2023, doi: 10.3390/logistics7010017.
[2] M. Mao and M. Hong, "YOLO object detection for real-time fabric defect inspection in the textile industry: a review of YOLOv1 to YOLOv11," *Sensors*, vol. 25, no. 7, Apr. 2025, doi: 10.3390/s25072270.
[3] G. Thilagavathi and T. Karthik, *Process control and yarn quality in spinning*, 1st ed. New York, United States: WPI Publishing, 2016, doi: 10.1201/b18797.

[4] J. Rodrigues, S. Anjos, M. Silva, R. Santos, and R. Oliveira, "Using soft computing and computer vision to create and control an integrated autonomous robotic manipulator process," in *Proceedings of the 26th International Conference on Enterprise Information Systems*, 2024, pp. 820–827, doi: 10.5220/0012705600003690.
[5] H. K. Kaul, "Applied sciences," in *Early Writings on India A Union Catalogue of Books on India in the English Language Published up to 1900 and Available in Delhi Libraries*, 1st ed., London, United Kingdom: Routledge, 2017, pp. 124–134, doi: 10.4324/9781315232140-14.
[6] A. Therese, "Robotic arm with real-time image processing using Raspberry Pi, both automated and manually," *International Journal of Advanced Research*, vol. 6, no. 3, pp. 1424–1430, Mar. 2018, doi: 10.21474/IJAR01/6815.
[7] M. S. B. and I. M. Makkubhai, "Four-DOF robotic arm system for goods transport," *International Journal For Multidisciplinary Research*, vol. 5, no. 4, Aug. 2023, doi: 10.36948/ijfmr.2023.v05i04.5946.
[8] R. N. Fauzan and R. Pramudita, "Controlling a 4 degree of freedom (4 DoF) robot arm with hand gestures using computer vision technology for manufacturing processes," *Journal of Electrical, Electronic, Information, and Communication Technology*, vol. 6, no. 2, pp. 48–54, Nov. 2024, doi: 10.20961/jeeict.6.2.90823.
[9] S. Vasudevan *et al.*, "Machine vision and robotics for primary food manipulation and packaging: a survey," *IEEE Access*, vol. 12, pp. 152579–152613, 2024, doi: 10.1109/ACCESS.2024.3479781.
[10] E. V. Filho, L. Lang, M. L. Aguiar, R. Antunes, N. Pereira, and P. D. Gaspar, "Computer vision as a tool to support quality control and robotic handling of fruit: a case study," *Applied Sciences*, vol. 14, no. 21, Oct. 2024, doi: 10.3390/app14219727.
[11] P.-S. Tsai, T.-F. Wu, J.-Y. Chen, and F.-H. Lee, "Drawing system with Dobot Magician manipulator based on image processing," *Machines*, vol. 9, no. 12, Nov. 2021, doi: 10.3390/machines9120302.
[12] P. Chand and S. Lal, "Vision-based detection and classification of used electronic parts," *Sensors*, vol. 22, no. 23, Nov. 2022, doi: 10.3390/s22239079.
[13] A. Mohamed *et al.*, "Design and control of a robotic manipulator for pick and place applications," *The International Undergraduate Research Conference*, vol. 6, no. 6, pp. 1–10, 2022, doi: 10.21608/iugrc.2022.302705.
[14] A. Iqdymat and G. Stamatescu, "Reinforcement learning of a six-DOF industrial manipulator for pick-and-place application using efficient control in warehouse management," *Sustainability*, vol. 17, no. 2, Jan. 2025, doi: 10.3390/su17020432.
[15] H. Sukri *et al.*, "Enhanced precision control of a 4-DOF robotic arm using numerical code recognition for automated object handling," *Journal of Robotics and Control*, vol. 6, no. 1, pp. 315–335, Feb. 2025, doi: 10.18196/jrc.v6i1.24349.
[16] E. A. Nugroho, J. D. Setiawan, and Munadi, "Image classification system design for yarn inspection process using teachable machine and Raspberry Pi," in *2024 IEEE International Conference on Smart Mechatronics (ICSMech)*, Nov. 2024, pp. 19–24, doi: 10.1109/ICSMech62936.2024.10812271.
[17] S. Cagnoni, M. Mordonini, L. Mussi, and G. Adorni, "Hybrid dual camera vision systems," in *Encyclopedia of Artificial Intelligence*, IGI Global, 2009, pp. 848–852, doi: 10.4018/978-1-59904-849-9.ch125.
[18] E. A. Nugroho, J. D. Setiawan, and M. Munadi, "Handling four DOF robot to move objects based on color and weight using fuzzy logic control," *Journal of Robotics and Control*, vol. 4, no. 6, pp. 769–779, Nov. 2023, doi: 10.18196/jrc.v4i6.20087.
[19] G. George, S. Adeshina, and M. M. Boukar, "Development of Android application for facial age group classification using TensorFlow Lite," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4, pp. 11–17, 2023.
[20] J. J. N. Wong and N. Fadzly, "Development of species recognition models using Google teachable machine on shorebirds and waterbirds," *Journal of Taibah University for Science*, vol. 16, no. 1, pp. 1096–1111, Dec. 2022, doi: 10.1080/16583655.2022.2143627.
[21] V. K. Mishra, S. Kumar, and N. Shukla, "Image acquisition and techniques to perform image acquisition," *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology*, vol. 9, no. 1, pp. 21–24, Jun. 2017, doi: 10.18090/samriddhi.v9i01.8333.
[22] R. Ponnusamy, S. Sathyamoorthy, and M. Kaliyamoorthi, "A review of image classification approaches and techniques," *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 3, pp. 1–5, Mar. 2017, doi: 10.23883/IJRTER.2017.3033.XTS7Z.
[23] S. Shawal, M. Shoyab, and S. Begum, "Fundamentals of digital image processing and basic concept of classification," *International Journal of Chemical and Process Engineering Research*, vol. 1, no. 6, pp. 98–108, 2014, doi: 10.18488/journal.65/2014.1.6/65.6.98.108.
[24] H.-L. Pham, V. Perdereau, B. V Adorno, and P. Fraisse, "Position and orientation control of robot manipulators using dual quaternion feedback," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 658–663, doi: 10.1109/IROS.2010.5651097.
[25] A. Sabo, H. Lábadi, T. Szakáll, and A. Kukla, "Controlling robotic arms using Raspberry Pi through the internet," in *International Conference on Applied Internet and Information Technologies*, Jun. 2016, pp. 171–180, doi: 10.20544/AIIT2016.21.
[26] R. Dasarwar, N. Thakre, K. Tikale, A. Dhoke, and T. Randive, "Industrial robotic arm using Raspberry Pi," *International Journal of Innovations in Engineering and Science*, vol. 9, no. 3, pp. 44–50, 2024, doi: 10.46335/IJIES.2024.9.3.9.
[27] C.-L. Shih, J.-H. Hsu, and C.-J. Chang, "Visual feedback balance control of a robot manipulator and ball-beam system," *Journal of Computer and Communications*, vol. 5, no. 9, pp. 8–18, 2017, doi: 10.4236/jcc.2017.59002.
[28] H.-R. Qu, J. Wang, L.-R. Lei, and W.-H. Su, "Computer vision-based robotic system framework for the real-time identification and grasping of oysters," *Applied Sciences*, vol. 15, no. 7, Apr. 2025, doi: 10.3390/app15073971.
[29] L. M. Silalahi, I. U. V. Simanjuntak, F. A. Silaban, S. Budiyanto, Heryanto, and M. Ikhsan, "Integration of OpenCV Raspberry Pi 3B+ and camera sensor in access control of vehicle ignition key system," in *IOP Conference Series: Materials Science and Engineering*, 2020, doi: 10.1088/1757-899X/909/1/012002.
[30] Ezenwobodo and S. Samuel, "An assessment on the use of mathematical softwares in teaching and learning of mathematics in colleges of education in South-Eastern Nigeria: a case study of Anambra and Enugu," *International Journal of Research Publication and Reviews*, vol. 4, no. 1, pp. 1806–1812, 2022, doi: 10.55248/gengpi.2023.4149.
[31] E. A. Nugroho, J. D. Setiawan, and M. Munadi, "Physical modeling of a 4-DOF manipulator for learning purposes using CAD and GUI," *International Journal of Mechanical Engineering and Robotics Research*, vol. 13, no. 3, pp. 392–401, 2024, doi: 10.18178/ijmerr.13.3.392-401.
[32] T.-Q. Nguyen *et al.*, "Kinematics, dynamics and control design for a 4-DOF robotic manipulator," in *2021 International Conference on System Science and Engineering (ICSSE)*, Aug. 2021, pp. 138–144, doi: 10.1109/ICSSE52999.2021.9538418.
[33] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, "Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 1376–1386, Apr. 2020, doi: 10.11591/ijece.v10i2.pp1376-1386.

BIOGRAPHIES OF AUTHORS

Emmanuel Agung Nugroho earned his Doctorate in Intelligent Mechatronics from Diponegoro University, Indonesia (2025), following a Magister Technic in Mechanical Engineering (2012) and a Bachelor of Electrical Engineering (2009). He is currently a lecturer at the Department of Mechatronics Engineering Technology, Politeknik Enjinering Indorama. His research focuses on electronic control systems, artificial intelligence, machine learning, and robotics. He has spoken at some reputable international conferences, published in many international reputable journals, and authored books, including machine learning: theory and application and mastering the fundamentals of mechatronics, showcasing his contributions to automation and intelligent systems. He can be contacted at email: emmanuel.agung@pei.ac.id.

Joga Dharma Setiawan received his B.Sc. from Northeastern State University, USA (1992), M.Sc. in Aeronautics and Astronautics from MIT (1996), and Ph.D. in Mechanical Engineering from Michigan State University (2000). He is a professor at Universitas Diponegoro and currently serves as President Director of PT LEN. Since 2019, he has led the Environmental Sensing Research Group under the MIT-Indonesia Research Alliance (MIRA). His research interests include robotics, IoT, nanosatellite tracking, electric vehicle control, and smart manufacturing. He actively collaborates with research centers such as CoREM, CBIOM3S, and NCSTT, and has published extensively in reputable journals and international conferences. He can be contacted at email: joga.setiawan@gmail.com.

Deni Kurnia holds a Bachelor of Education in Electrical Engineering from the Universitas Pendidikan Indonesia and a Master of Engineering in Electrical Engineering from the Bandung Institute of Technology (ITB). Currently, he is pursuing his doctoral degree in Mechanical Engineering at Universitas Andalas under a research fellowship from the National Research and Innovation Agency (BRIN) through the degree by research program. He serves as a lecturer in Mechatronics Engineering at the Politeknik Enjinering Indorama, Purwakarta, Indonesia. His research focuses on IoT, machine learning, and smart factory applications. He has been awarded multiple competitive research grants from the Indonesian Ministry of Research, Technology, and Higher Education. He can be contacted at email: deni.kurnia@pei.ac.id.

Nanang Roni Wibowo holds a Bachelor of Engineering (B.Eng.) in Electrical and Electronics Engineering from National Institut of Technology Malang, Master of Engineering (M.Eng.) in Electrical and Electronics Engineering from Hasanuddin University, besides several professional certificates and skills. He is currently lecturing with the Department of Mechatronics Engineering Technology at Politeknik Enjinering Indorama, Purwakarta, West Java Province, Indonesia. He is a member of the Indonesian Association of Automation and Mechatronics Engineer (IKATOMI). His research areas of interest include automatic control, artificial intelligent, digital signal processing, and embedded system. He can be contacted at email: nanang.roni@pei.ac.id.