Dang Phuoc Thinh
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Van Thi Ngoc Han
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Nguyen Nghiep Thinh
Bui Thi Xuan High School,73-75 Bui Thi Xuan Street, Ben Thanh Ward, Ho Chi Minh City Vietnam

Nguyen Vu Duc Huy
Le Quy Don High School for the Gifted, 3/2 Street, Phuoc Thang Ward, Ho Chi Minh City
Vietnam

Tran Quang Thuan
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Huynh Nhat Duy
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Phi Thanh Dat
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Vo Cong Hieu
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Vo Duy Cong
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Development of an Educational Robotic Platform with Dual Control Interfaces and Intelligent Computer Vision-Based Interaction

This paper presents the development of a low-cost, AI-enabled educational robotic platform designed to enhance hands-on STEM learning. The system consists of a 4-degree-of-freedom (DOF) robotic arm constructed from 3D-printed components, controlled by an ESP32-based circuit and integrated with both local and web-based interfaces for flexible operation. Two control interfaces were developed: a local desktop GUI built with PyQt5 and a remote web-based interface using the MQTT protocol. Both interfaces allow users to manually control robot joint angles, adjust movement step sizes, reset to default positions, and monitor real-time joint states. These interfaces provide intuitive interaction, enabling students to understand motion control in robotics. To integrate artificial intelligence and computer vision, three modules were implemented: face tracking, hand gesture control, and object detection. Face tracking translates facial position and size into 3D coordinates for real-time robot movement using inverse kinematics. Hand gesture recognition uses MediaPipe to interpret finger poses and execute corresponding robot actions. The object detection module employs a YOLOv12 model trained on classroom objects (pens, erasers, markers) to perform autonomous pick-and-place tasks on a simulated conveyor system. Experimental results validate the system's effectiveness in real-time tracking, gesture interpretation, and object manipulation. Real-time plots of joint angles and workspace coordinates illustrate the system's responsive behavior. This integrated platform enables students to explore AI, vision, and robotics in a unified environment. Through interactive exercises, they gain practical experience in programming, AI model development, user interface design, and robotic control, bridging theoretical knowledge and real-world applications.

Keywords: *Educational Robotics; STEM Education; Robotic Arm; Artificial Intelligence; IoT.*

1. INTRODUCTION

STEM (Science, Technology, Engineering, and Mathe–matics)education has emerged as a critical pillar in preparing future generations for the demands of an increasingly technology-driven world. By emphasizing the integration of Science, Technology, Engineering, and Mathematics, STEM curricula aim to cultivate problem-solving abilities, computational thinking, design creativity, and interdisciplinary collaboration among students [1-3]. The growing complexity of mo–dern engineering and automation systems underscores the need for students to engage with real-world tools and scenarios from an early stage. Numerous studies have demonstrated that active, project-based learning approaches significantly enhance the retention of know–ledge and the development of technical competencies, making STEM education an essential component of national and global educational strategies [4-5]. As the Fourth Industrial Revolution unfolds, competencies in fields such as artificial intelligence (AI), computer vi–sion, and the Internet of Things (IoT) have become indispensable across industries [6-7]. Consequently, there is a growing need to modernize STEM education by integrating these cutting-edge technologies into hands-on, project-based learning environments that fos–ter both technical knowledge and system-level thinking.

Robotics has proven to be one of the most powerful tools for implementing STEM education in practice. Through robotics, learners can gain hands-on experi–ence in mechanical design, electronics, programming, sensor integration, and system-level thinking. The interactive and visual nature of robotic systems not only makes abstract engineering concepts more accessible but also increases student engagement, particularly in K–12 and early undergraduate education. Robotic sys–

Received: September 2025, Accepted: December 2025
Correspondence to: Vo Duy Cong, Faculty of Mechanical Engineering,Ho Chi Minh City University of Technology (HCMUT), Vietnam
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2601093P

tems offer learners the opportunity to design, build, test, and refine solutions to real-world challenges, while developing essential 21st-century skills such as teamwork, innovation, and resilience.

Over the past decade, a wide range of educational robotic platforms have been developed and deployed with the aim of supporting STEM education. These systems not only promote hands-on learning but also bridge theoretical knowledge with practical applications through interdisciplinary activities. In [8], Mondada et al. presented R2T2, a cross-continental collaborative robotic event engaging over 100 students from Europe and Africa in a space rescue scenario using real-time video streaming and remote programming. This initiative emphasized pedagogical outcomes over competition, highlighting inclusion and the adoption of enhanced methodologies among African participants. In [9], Zeng et al. developed iArm, a 6-DOF robotic arm kit with vision and conveyor modules, aiming to enhance students' computational thinking through a semester-long program. Their study showed notable improvements in students' abilities, demonstrating the educational impact of structured robotics curricula. Robotic platforms have also proven effective for special education. For instance, Bargagna et al. in [10] adapted educational robotics for children with Down syndrome, showing how the Bee-Bot robot can support cognitive and social development in inclusive settings.

Mobile robotics has gained attention for its accessibility and versatility. In [11], Nguyen et al. integrated a RockChip AI processor with voice recognition (Vietnamese, English, Korean) into a mobile robot controlled via Android, enabling multilingual STEM learning through natural interaction. Similarly, Haruna et al. in [12] developed a mobile robot for teaching material handling in mechanical workshops, demonstrating improved learning outcomes through quasi-experimental evaluation. Expanding affordability and customization, Pedre et al. introduced ExaBot [13], a multi-purpose, open-source mobile robot that is more than ten times less expensive than commercial platforms, serving both research and outreach purposes. Desktop manipulators also gained traction, with Vega and Pérez [14] proposing G-ARM, a low-cost, ROS2-integrated robotic arm validated in university courses for motion planning and simulation.

Several studies focus on lowering the cost barrier while maintaining functionality. In [15], Eaton and Tanveer designed a cost-effective ROS-based robotic arm for instructional labs, offering insights into component selection and enabling broader student participation. Čehovin Zajc et al. [16] presented a low-weight, open-source robotic platform that was validated across different age groups and academic levels, confirming its motivational and cognitive benefits. Meanwhile, Chavdarov et al. [17] proposed a 3D-printed Delta robot for educational purposes, emphasizing ease of manufacturing, geometric kinematics, and trajectory accuracy.

More recently, integrating AI and IoT into educational robotic platforms has opened new opportunities to enhance student engagement and expose learners to intelligent and connected systems — capabilities that are becoming increasingly essential in smart manufacturing, service robotics, and cyber-physical systems. While existing robotic educational platforms provide meaningful engagement in mechanics and control systems, most are limited in their integration of emerging technologies such as artificial intelligence (AI), computer vision, and the Internet of Things (IoT). These technologies are not only transforming modern industry but are also becoming increasingly relevant in educational contexts, enabling students to explore intelligent automation, human–robot interaction, and cloud-based control systems.

Most available educational robotic arms typically support only basic kinematic control and rely on scripted motions or button-based interfaces. Although some systems allow for sensor integration or mobile app control, they rarely incorporate advanced features like deep learning-based object recognition, vision-guided manipulation, or real-time remote interaction through IoT protocols. As a result, they fail to provide students with a holistic understanding of modern intelligent robotic systems.

Several recent efforts have attempted to bridge this gap. For example, Vega and Cañas [18] introduced PiBot, an open, low-cost mobile robot built on Raspberry Pi, featuring an onboard camera and a Python-based software infrastructure. While it enhances visual learning through onboard vision, its application is limited to basic perception and does not yet incorporate AI or IoT functionality. Similarly, Zheng [19] proposed a robot interaction system utilizing computer vision and deep machine learning to support English cultural education; however, the application is domain-specific and not oriented toward general robotics control or STEM-focused platforms. In another study, Antonio et al. [20] presented a mechatronics training system that utilizes computer vision for 2D positioning of LEGO mobile robots. Although effective in teaching fundamental robotics concepts, the system lacks integration with AI-based perception or remote connectivity. In [21], Mamatnabiyev et al. proposed FOSSBot, an open-source robot designed to support a wide range of computer science courses in tertiary education. A major strength of this platform lies in its broad applicability, as it is equipped with IoT technologies for sensing and control, and supports courses in AI, computer vision, electronics, and networking. Its open hardware/software architecture allows deep customization and curriculum alignment. However, despite its flexibility, FOSSBot is primarily targeted at software-level educational activities and lacks physical manipulation capabilities, such as robotic arms or grippers, which limits its applicability in teaching mechanical design or controlling articulated robots. Manzoor et al. [22] developed a 6-DOF articulated robotic arm equipped with an onboard camera and force sensor at the end effector, enabling autonomous object manipulation and advanced control experiments. This platform is particularly strong in supporting education in mechanical engineering, robotics, and control systems, with applications ranging from grasping to trajectory planning. Moreover, its open-source nature and rich sensory feedback allow both beginners and advanced users to explore a wide spectrum of robotics applications. Nevertheless, while the system supports basic perception through vision, it does not natively integrate modern AI-based object recognition or IoT-based remote access, which limits its alignment with current trends in intelligent robotics.

Educational robots that integrate advanced features —such as AI-enhanced vision modules or IoT connec–tivity—are often prohibitively expensive or difficult to customize because of closed hardware/ software eco–systems. This situation creates a critical gap in STEM education: there is still a lack of affordable, open, and extensible robotic platforms that simultane–ously support mechanical design learning, embedded control, AI-driven perception, and IoT-based communi–cation within a unified framework.

To address this gap, this paper presents a novel, low-cost, and fully integrated 4-DOF educational robotic arm platform intended for hands-on STEM and mecha–tronics training. The proposed system consists of four tightly coupled subsystems: (i) a servo-driven robotic manipulator with a gripper, (ii) an ESP32-based embed–ded control unit enabling I2C communication for multi-channel PWM generation, (iii) dual control interfaces that support both local operation (desktop PyQt5 GUI) and remote access (web/IoT-based control), and (iv) an AI-powered computer vision module (YOLO-based) for object detection and vision-based interaction.

The platform provides a reproducible reference de–sign that can be directly deployed in teaching labora–tories to support experiments in robot kinematics, actuator control, embedded programming, sensor/ communication integration, and AI-based perception. By combining low-cost hardware with an open and modular architecture, the system can be easily custo–mized for different curricula and extended with addi–tional sensors or algorithms, thereby lowering the entry barrier for institutions and enabling scalable, practice-oriented robotics education.

The remainder of this paper is organized as follows. Section 2 presents an overview of the overall system architecture and its key components. Section 3 details the design of the robotic arm system, including mecha–nical construction, control circuitry, human–machine interface, and kinematic modeling. Section 4 focuses on the integration of artificial intelligence and computer vision, highlighting applications such as face tracking, hand gesture recognition, and object detection for pick-and-place tasks. Section 5 presents and discusses the experimental results, demonstrating the educational po–tential and technical performance of the system. Finally, Section 6 concludes the paper and outlines future directions for enhancement and classroom deployment.

2. SYSTEM ARCHITECTURE

The proposed robotic arm platform is designed as a comprehensive educational tool that integrates mecha–nical design, embedded electronics, IoT connectivity, and artificial intelligence to support hands-on STEM learning. The system is composed of four tightly integ–rated subsystems: the mechanical structure, the elect–ronic control unit, the user interfaces (both local and remote), and the AI-powered computer vision module. Figure 1 illustrates the overall system architecture.

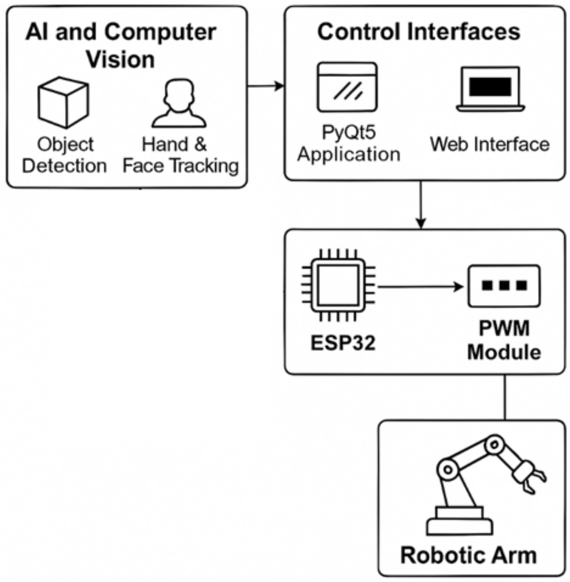


Figure 1. System architecture of the robotic arm platform for STEM education, integrating AI-based vision, IoT communication, and multimodal control interfaces

At the core of the system lies a 4-DOF robotic arm, which is actuated using multiple MG996R servo mo–tors. The robot is designed using Autodesk Inventor and fabricated with 3D-printed components, enabling cost-effective manufacturing and easy maintenance. The end-effector is a servo-actuated gripper, capable of per–forming basic manipulation tasks such as object grasping and placement.

The control unit is built around the ESP32 micro–controller, which provides both computation and wire–less connectivity. A PCA9685 PWM driver is used to generate the PWM signals required to control up to 16 servo motors, communicating with the ESP32 via the I2C protocol. The entire circuitry is custom-designed and documented using Altium Designer, ensuring reliable operation and reproducibility for educational deployments.

To interact with the robotic arm, two types of control interfaces are provided:

- A desktop application built with PyQt5, allo–wing users to control each joint of the robot locally and visualize joint angles in real-time.
- A web-based control interface developed using the Blynk IoT platform, which enables users to remotely operate the robot through a smartphone or tablet, illus–trating key concepts in IoT and remote robotics.

Additionally, the platform is enhanced with an AI and computer vision module. A camera mounted on or near the robot captures real-time video, which is then processed using pre-trained and custom-trained deep learning models, YOLOv12 for object detection. The system also supports gesture-based control and human-robot interaction by leveraging models capable of hand tracking and face detection, empowering students to explore applications of AI in robotics.

This robotic platform is purposefully designed to serve as a multidisciplinary learning tool, supporting a wide range of topics including mechanical design, electronic circuit development, embedded systems programming, IoT communication, AI model deploy–

ment, and computer vision applications. It provides students with an integrated environment that allows them to practice both theoretical and practical aspects of STEM disciplines in a hands-on and engaging manner.

3. DESIGN OF THE ROBOT ARM SYSTEM

3.1 Mechanical design

Figure 2 illustrates the complete mechanical structure of the robotic platform, which consists of a low-cost, 3D-printed, 4-degree-of-freedom (DOF) manipulator integ–rated with a conveyor belt system. Designed primarily for educational and research purposes, the platform de–monstrates practical applications of robotics, automa–tion, and intelligent systems. The manipulator is equipped with a servo-actuated gripper at the end-effector, enabling autonomous grasping and arrange–ment of objects transported by the conveyor. The system is designed to be compatible with AI and com–puter vision modules, allowing for the implementation of object detection, tracking, and intelligent manipu–lation tasks in real-world scenarios.

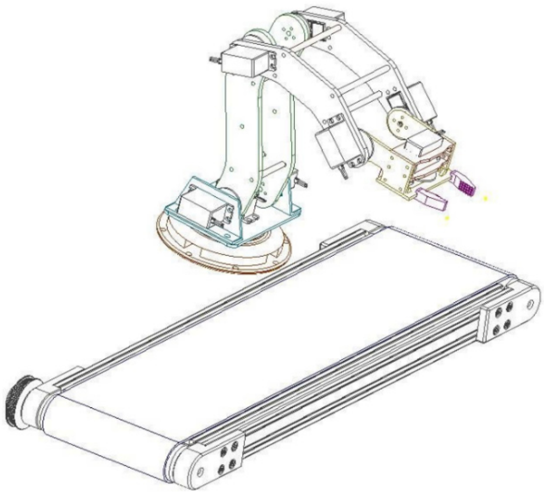


Figure 2. Diagram of the complete mechanical system

Figure 3. The manipulator design using Autodesk Inventor

Figure 3 illustrates the 3D design of the robot arm modeled in Autodesk Inventor. The robot arm features 4-DOF, with each joint actuated by an MG996R servo motor, offering precise motion control at an affordable cost, making it suitable for educational applications. The specifications of the servo motors are provided in Table 1. Since each servo delivers a relatively low tor–que of 11 kg·cm, the 2nd, 3rd, and 4th joints are equip–ped with dual motors to ensure sufficient torque for rotating the corresponding links.

The robotic manipulator possesses 4 DOF. The first joint enables rotation about the vertical (Z) axis, while the remaining joints provide motion around horizontal axes. Although each MG996R servo motor is capable of rotating up to 180 degrees, the actual range of motion at each joint is constrained by the mechanical structure of the system. The specific angular limits for each joint are summarized in Table 2.

Table 1. Specifications of the MG996R

Specification	Value
Nominal voltage	4.8 to 6V
Speed	4.8V: 0.19s/60-degree rotation 6.0V: 0.15s/60-degree rotation
Current draw	10mA idling 170mA operating without a load 1400mA at stall
Stall torque	4.8V: 9.4kg/cm 6.0V: 11kg/cm
Dimensions	40.7x19.7x42.9mm

Table 2. Joint Rotation Ranges of the 4-DOF Robotic Manipulator

Joint	Range
The first joint	0 to 180 degrees
The second joint	20 to 150 degrees
The third joint	-60 to 60 degrees
The fourth joint	-90 to 90 degrees
The gripper	0 to 9 degrees

As shown in Figure 4, the base of the manipulator consists of a rotating platform and a servo mounting bracket. A centrally mounted servo motor provides rotation around the vertical (Z) axis, enabling the first degree of freedom. To support the lifting of the arm structure, two additional servo motors are symmetrically mounted on either side of the base. These servos operate in parallel to actuate the second joint, thereby increasing the available torque and ensuring stable motion of the upper links.

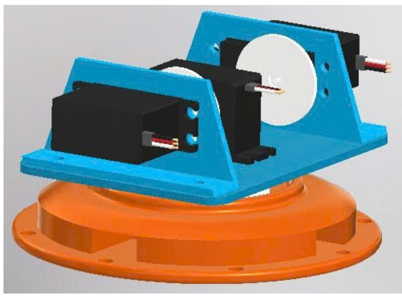


Figure 4. The Base of the Manipulator

As illustrated in Figure 5, the first link serves as the intermediary component connecting the second and third degrees of freedom. The lower mounting holes are designed to interface with the actuators of the second joint, while the upper holes connect to the third joint's actuators. The link consists of two S-shaped plates, each 5 mm thick. When assembled, the component measures 180 mm in length and 60 mm in width.

The second link incorporates a servo mounting structure that is nearly identical to the first link's design, except for an additional 5 mm in width when assem–bled. This component mounts four servo motors: two are aligned with the upper mounting holes of the first link to actuate the third degree of freedom, while the remaining two drive the end-effector, enabling the fourth degree of freedom (see Fig. 6).

Figure 5. Structural Design of the First Link in the Robotic Manipulator

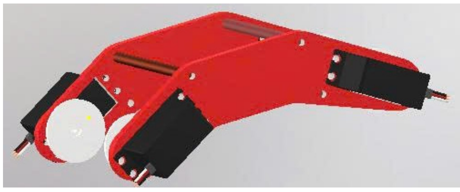


Figure 6. Servo Mounting Configuration for the Third and Fourth Degrees of Freedom

The end effector, as shown in Figure 7, is a 3D-printed component designed to achieve cost-effecti–veness and facilitate easy replication. It consists of two plates with mounting holes that connect to the pair of servo motors responsible for actuating the fourth joint. An additional servo motor is integrated to control the fifth degree of freedom, which drives a two-finger gripper mechanism used for object manipulation

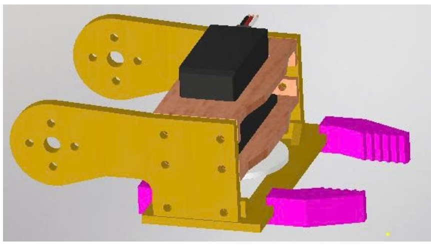


Figure. 7. The end effector

3.2 Control circuit design

Figure 8 illustrates the overall architecture of the control circuit for the robotic platform. The system integrates key electronic components, including an ESP32 micro–controller, a PCA9685 16-channel PWM driver, a relay module, a 12V power supply, and a 5V voltage regu–lator module. Each component is functionally assigned to ensure the reliable and coordinated operation of the robotic manipulator and the conveyor system.

The ESP32 acts as the central processing unit of the control system. It is responsible for communicating with the user interface—either through a local PC connection or a cloud-based web interface—allowing real-time wireless control and monitoring. Upon receiving control commands, the ESP32 transmits the corresponding PWM control signals via the I²C protocol to the PCA9685 module, which precisely regulates the rota–tion angles of the eight servo motors responsible for actuating the five degrees of freedom of the robotic arm. These include dual-servo configurations at certain joints to ensure sufficient torque.

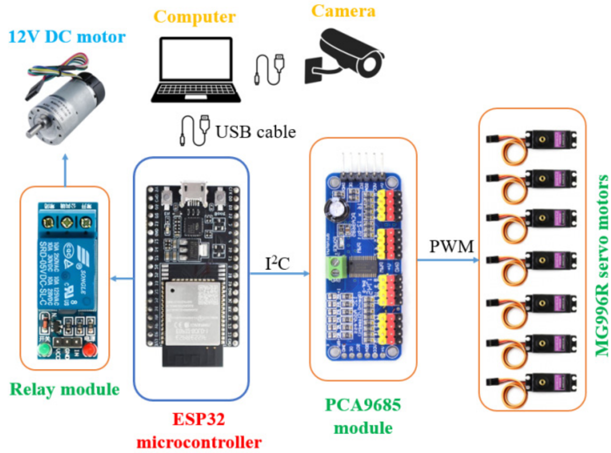


Figure 8. The overall architecture of the control circuit

The PCA9685 module is a 16-channel, 12-bit PWM driver that communicates with the ESP32 using the I²C protocol, allowing for the simultaneous control of up to 16 servo motors using only two GPIO pins. In this system, it controls eight MG996R servo motors for the 4-DOF robotic arm. The module operates at 5V and supports PWM frequencies from 24 Hz to 1526 Hz, with a 12-bit resolution (4096 steps), ensuring smooth and accurate servo movement. An external power input enables the module to supply sufficient current to all connected servos, enhancing stability and reducing load on the ESP32.

In parallel, the ESP32 also outputs a digital control signal to a 5V relay module, which in turn switches a 12V DC motor responsible for driving the conveyor belt. This indirect control via relay ensures isolation between the logic and power circuits, improving both safety and durability.

Power for the system is supplied by a 12V, 5A adapter with a 220V AC input, providing sufficient current for both the servos and the conveyor motor. A DC-DC converter module is employed to step down the 12V supply to 5V, which powers the PCA9685 and other low-voltage peripherals.

3.3 Control Interface Design

To enable flexible and user-friendly interaction with the robotic platform, two control interfaces were developed: one for local control via USB connection to a computer, and another for remote control over the Internet. Both interfaces are designed to send control commands to the ESP32 microcontroller, which interprets the input and actuates the corresponding joints of the robot. The local interface is implemented as a desktop application for direct, real-time control during development and testing, while the web-based interface allows users to operate the system remotely using mobile or cloud-connected devices.

A desktop-based control interface was developed to enable real-time and intuitive manipulation of the robot arm. As illustrated in Figure 8, the interface was designed using Qt Designer and implemented in Python using the PyQt5 library. It provides individual control panels for each joint, including the base, shoulder, elbow, forearm, wrist, and gripper.

Each panel features a slider that enables users to quickly adjust the corresponding joint to any desired position. Radio buttons are provided to select the step size (C1–C5), which determines the number of degrees the joint angle changes when the Increase or Decrease buttons are pressed. Smaller step values are used for fine adjustments, while larger steps enable faster move–ments. A Reset button is included to return the robot to its default home position, with joint angles preset to [0°, 90°, 90°, 0°, 0°].

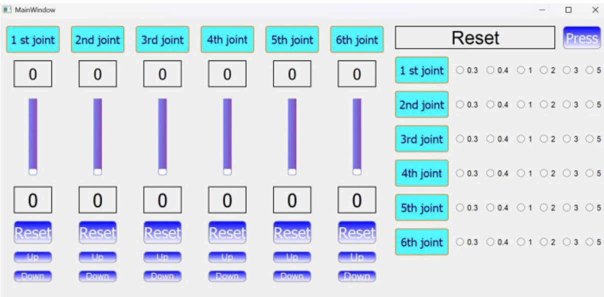


Figure 9. Desktop-Based Graphical User Interface for Robotic Arm Control

Additionally, each joint panel includes a label displaying the current angle in real-time, allowing the user to monitor joint positions throughout operation. This interface offers a flexible and user-friendly envi–ronment for both precise joint tuning and rapid pose adjustment, making it particularly effective for educa–tional demonstrations and development tasks.

In addition to the desktop application, a web-based control interface was developed to enable remote opera–tion of the robotic system over the Internet. As shown in Figure 9, this interface replicates the functionality of the local GUI, including joint-specific sliders, current angle displays, step adjustment options (via radio buttons), and reset functions.

Figure 10. Web-Based Control Interface Using MQTT Protocol for Remote Robot Operation

The web interface communicates with the ESP32 microcontroller using the MQTT (Message Queuing Telemetry Transport) protocol, a lightweight and well-suited protocol for real-time IoT applications. Com–mands issued from the web interface, such as adjusting joint angles or initiating a reset, are published as MQTT messages and received by the ESP32, which then performs the corresponding control actions.

This web-based interface provides the flexibility to remotely control the robot from any device with Internet access, making it particularly useful for educational de–monstrations, distributed learning environments, or applications where physical proximity to the robot is not feasible.

3.4 Robot arm kinematics

The robotic arm features a 4-DOF configuration for spatial positioning and an additional DOF for end-effector actuation. Figure 11 presents the kinematic structure of the robotic arm, including the joint angles θ_0, θ_1, θ_2, θ_3, link lengths l_1, l_2, l_3, and the vertical offset (d) from the base. To control the robot in Cartesian space, both forward and inverse kinematics are formu–lated to compute the mapping between joint angles and end-effector coordinates.

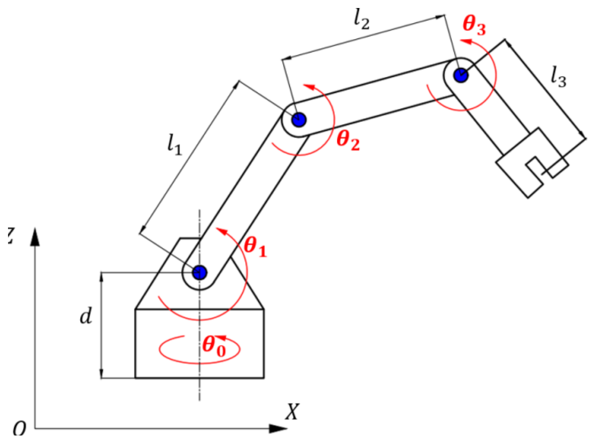


Figure11. Kinematic diagram of the robotic arm

The position of the end-effector (x, y, z) in the world coordinate frame can be expressed as a function of the joint angles (θ_0, θ_1, θ_2, θ_3) and the link lengths l_1, l_2, l_3, as follows:

$$\begin{aligned} x &= \left(l_1 c_1 + l_2 c_{12} + l_3 c_{123}\right) c_0 \\ y &= \left(l_1 c_1 + l_2 c_{12} + l_3 c_{123}\right) s_0 \\ z &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} + d \\ \theta &= \theta_1 + \theta_2 + \theta_3 \end{aligned} \tag{1}$$

where, $c_0 = \cos\theta_0$, $s_0 = \sin\theta_0$, $c_1 = \cos\theta_1$, $s_1 = \sin\theta_1$, $c_{12} = \cos(\theta_1 + \theta_2)$, $s_{12} = \sin(\theta_1 + \theta_2)$, $c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, $s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$, θ_0 represents the base rotation, while θ_1, θ_2, θ_3 correspond to the vertical plane joint angles.

Given a target position (x, y, z), the joint angles can be computed by solving the inverse kinematics as follows:

First, compute the base rotation angle:

$$\theta_0 = a\tan 2\left(y.x\right) \tag{2}$$

Sum the squares of the two sides of Equation (1):

$$\begin{aligned} l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) &= \sqrt{x^2+y^2} - l_3\cos\theta \\ l_2\sin\theta_1 + l_2\sin(\theta_1+\theta_2) &= z - d - l_3\cos\theta \end{aligned} \tag{3}$$

Denote $m = \sqrt{x^2+y^2} - l_3\cos\theta$, $n = z - d - l_3\cos\theta$. Sum of the squares of the two sides of the Equation (3):

$$l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2 = m^2 + n^2 \tag{4}$$

So we can solve θ_2:

$$\cos\theta_2 = \frac{m^2+n^2-l_1^2-l_2^2}{2l_1l_2} \tag{5}$$

Denote $k_1 = (l_1 + l_2\cos\theta_2)$, $k_2 = l_2\sin\theta_2$. Transform Equation (3) to get:

$$\begin{aligned} k_1\cos\theta_1 - k_2\,\mathrm{sins}\,\theta_1 &= m \\ k_1\sin\theta_1 + k_2\cos\theta_1 &= n \end{aligned} \tag{6}$$

Solve for θ_1:

$$\theta_1 = a\tan 2(m,n) - a\tan 2(k_1,k_2) \tag{7}$$

Finally, determine θ_3:

$$\theta_3 = \theta - \theta_1 - \theta_2 \tag{8}$$

This inverse kinematics formulation enables the robot to compute the necessary joint angles to reach a desired target position, which is essential for executing robot control.

4. AI AND COMPUTER VISION SYSTEM

4.1 Face tracking and hand gesture control

To enhance the intuitiveness and interactivity of the robotic platform, a control mechanism based on facial tracking and hand gesture recognition was implemented using AI-based computer vision techniques. This app–roach enables users to operate the robot without requi–ring physical interfaces, making the system particularly suitable for educational environments and human-robot interaction research.

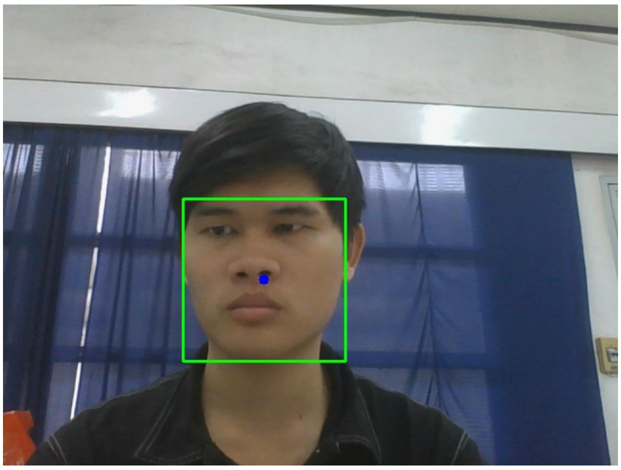


Figure 12. Face detection using the MediaPipe Face Detection model

The face tracking system, as illustrated in Figure 12, employs the MediaPipe Face Detection model [23], which performs real-time localization of the user's face using a lightweight convolutional neural network. Once a face is detected, the algorithm provides a bounding box from which the center coordinates (u, v) and the area s of the face are computed. Specifically, (u, v) denote the pixel coordinates of the bounding-box centroid in the image plane, where u and v correspond to the horizontal (x-axis) and vertical (y-axis) directions, respectively. The face area s is defined as the bounding-box area in pixels, calculated as $s = height \times width$ (in pixels^2).

These parameters are then mapped to the robot's spatial coordinates (X, Y, Z) based on the following transformation equations:

$$\begin{aligned} Y &= \left(u - \frac{w}{2}\right)k_y \\ Z &= -\left(v - \frac{h}{2}\right)k_z + l_1 + d \\ X &= sk_x + l_2 \end{aligned} \tag{9}$$

where w and h are the width and height of the video frame, and k_x, k_y and k_z are scaling factors that adjust the sensitivity and fit the robot's workspace. This mapping enables the robot to track the user's facial position in three dimensions (horizontally, vertically, and in depth) based on simple head movements and the distance from the camera.

To recognize hand gestures, the system utilizes MediaPipe Hands[24-25], a deep learning-based model that detects and tracks 21 key landmarks on the human hand in real-time. These landmarks represent anato–mical features such as finger tips, joints, and the wrist, enabling the system to analyze the spatial configuration of the hand with high precision, as shown in Figure 13.

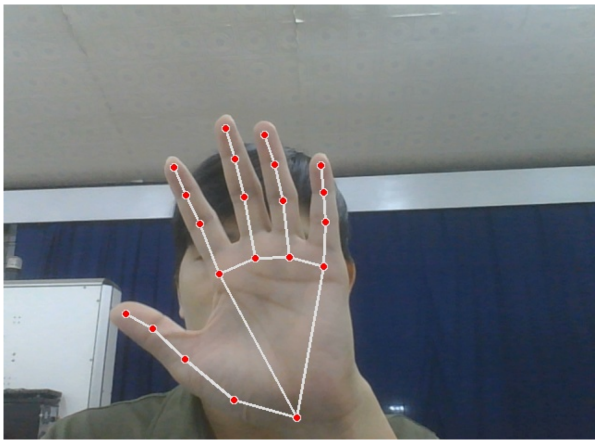


Figure 13. Hand gestures detection with 21 landmarks

The detection of gesture states is based on two key aspects: (1) the number of extended fingers, determined by comparing the position of each fingertip to its corresponding joint. (2) The orientation of the palm, inferred from the relative positions of key landmarks (e.g., wrist vs. fingertips), to determine whether the palm is facing up or down.

To determine which fingers are extended, the system compares the vertical (Y-axis) coordinates of each fingertip to the joint two levels below it:

- If the Y-coordinate of the fingertip is smaller (i.e., higher in the image frame) than the joint below it, the finger is considered extended.
- Otherwise, it is classified as folded.

For the thumb, which is oriented laterally, the X-axis is used instead:

- On the right hand, the thumb is extended if the tip is to the left of its joint.
- On the left hand, the thumb is extended if the tip is to the right of its joint.

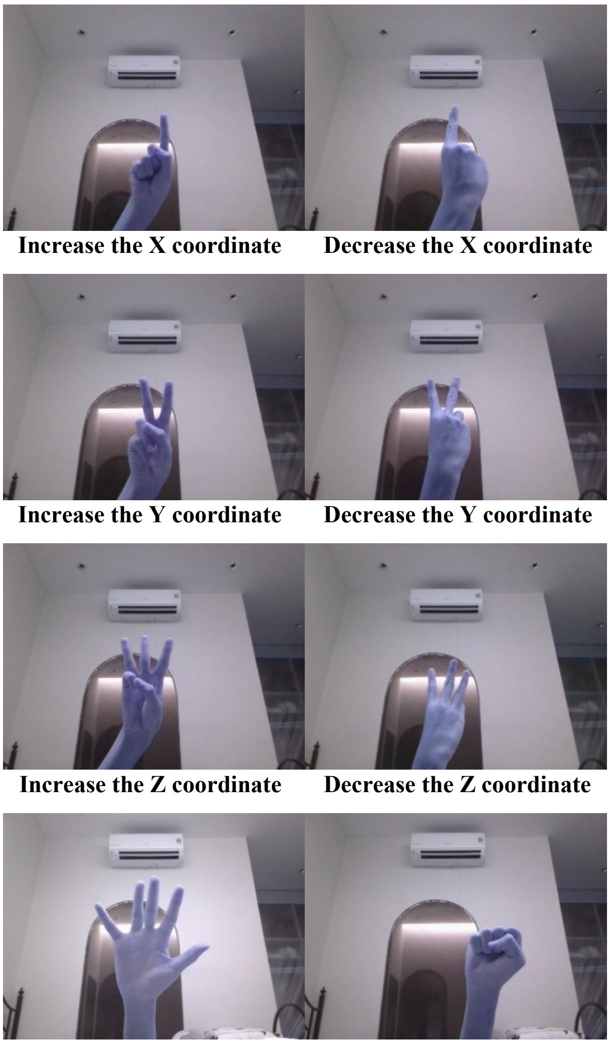
Once the system identifies which fingers are extended, it counts the number of raised fingers and evaluates the overall hand configuration. The gesture classification rules are then applied as follows (see Figure 14):

Raising one finger with the palm facing up or down to increase or decrease the robot's X-coordinate, respectively.

Raising two fingers (index and middle) with the palm facing up/down to control the Y-coordinate.

Raising three fingers (index, middle, and ring) with the palm facing up/down to adjust the Z-coordinate.

A fully open palm is interpreted as a command to open the gripper, while a closed fist is used to close the gripper.



Open gripper Close gripper

Figure 14. The gesture classification rules

This multimodal control strategy enables manipulation of the robot's position and end-effector in a contactless manner. The combination of facial and hand gesture inputs provides both gross and fine control capabilities, enhancing the user experience and demonstrating the potential of AI-driven human-robot interaction.

4.2 Object Detection for Robot Pick-and-Place Applications

To enable autonomous pick-and-place functionality, an object detection system was integrated into the robotic platform. The goal is to allow the robot to recognize and manipulate commonly used classroom items such as pencils,markers, and erasers (see Figure 15). These objects are typically small, lightweight, and varied in shape and color—posing a moderate challenge for vision-based systems in real-world conditions.

Figure 15. Objects in the dataset

To address this, we trained a custom object detection model based on YOLOv12, a state-of-the-art deep learning architecture known for its balance between detection accuracy and inference speed [26]. The training dataset was collected specifically for this application, comprising approximately 200 images per object class. Images were captured under different lighting conditions and backgrounds to improve the model's generalization capability. The annotation process was carried out using the Roboflow platform [27], which provided a streamlined workflow for labeling and dataset augmentation.

Model training was conducted on Google Colab using GPU acceleration. The training process involved multiple epochs with early stopping and validation monitoring to prevent overfitting. Upon achieving satisfactory accuracy and loss convergence, the trained model was exported to the deployment format compatible with the robot's control system.

During real-time operation, the YOLOv12 model is executed on a local computing unit connected to the vision system. Once an object is detected, the bounding box coordinates are extracted and processed to compute the object's position relative to the camera. This 2D information is then mapped to the robot's workspace through a calibrated transformation, allowing the system to calculate the corresponding joint angles required to reach the object. These angles are subsequently sent to the servo controller, enabling the robotic arm to execute the pick-and-place task.

The primary purpose of this application is to provide students with practical experience in applying artificial intelligence to real-world robotics problems. Through this task, students learn how to collect and annotate image data, train and evaluate object detection models, and deploy them within a functioning robot system. Furthermore, they gain insight into how vision-based outputs are converted into spatial positions and then into joint angle commands to control the robot. This hands-on activity helps bridge the gap between theoretical concepts and real-world applications, fostering inter–disciplinary skills in AI, computer vision, control engi–neering, and automation.

5. RESULTS AND DISCUSSIONS

The robotic arm was fabricated using 3D printing technology, enabling the creation of low-cost, modular, and easily replicable components. After printing, all joints and links were manually assembled into a comp–lete manipulator model as shown in Figure 16. During operation, the robotic arm is connected to a laptop that runs both the control interfaces and the image pro–cessing programs required for various tasks. These programs are optimized to run efficiently on standard laptops equipped with only a CPU, without the need for expensive GPU hardware. This design choice ensures accessibility and cost-effectiveness for educational en–vironments. The vision system can utilize either the laptop's built-in webcam or an external USB camera, offering flexibility depending on the available hardware. This setup allows students to easily experiment with AI-driven robotic applications using readily available and affordable equipment.

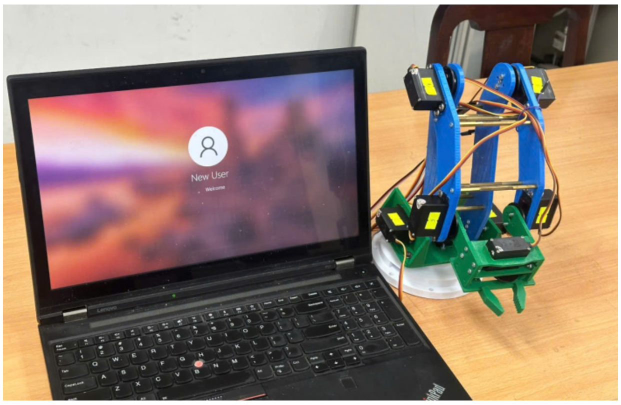


Figure 16. The 3D-printed robotic arm is connected to a laptop running control and vision-based applications.

To evaluate the face tracking functionality, the sys–tem was tested under real-time conditions using a webcam as the input source. The results are illustrated in Figure 17, which shows the detected face enclosed in a green bounding box, with the center point (u, v) marked, and the estimated bounding box area s. These values were used to compute the corresponding 3D coordinates (X, Y, Z) of the target position using a calibrated linear mapping.

Additionally, to monitor the robot's response to the tracked target, the end-effector coordinates and cal–culated joint angles were plotted over time, as shown in Figure 18. The upper graph demonstrates the dynamic changes in spatial position X, Y, Z while the lower graph visualizes the angular displacement of each joint (θ_0, θ_1, θ_2, θ_3) in real time. The plotted data confirm that the robot successfully followed the user’s head movement, making continuous adjustments to its joints based on the inverse kinematics solution.

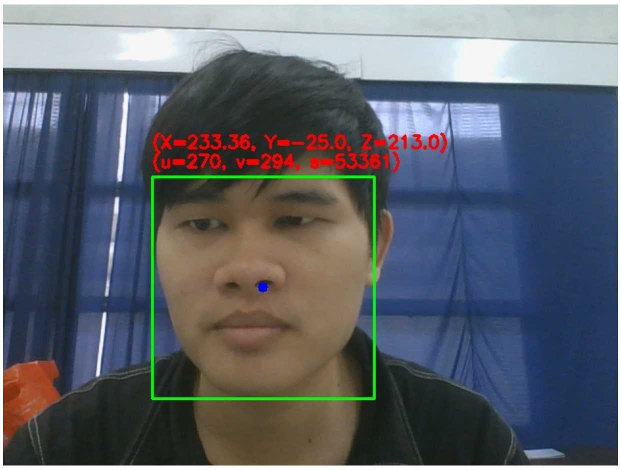


Figure 17. Result of the face tracking module.

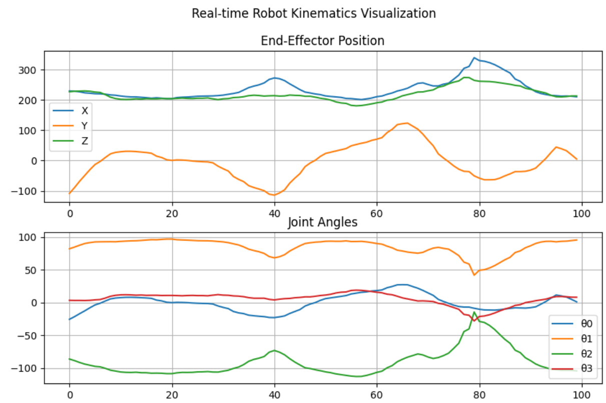


Figure 18. Real-time plots showing the robot’s kinematic response.

To evaluate the effectiveness of gesture-based cont–rol, a series of predefined hand poses was tested using the MediaPipe Hands module. The system was configured to recognize eight distinct gestures, each corresponding to a specific control command for the robot's motion or gripper operation. The results of the gesture recognition are illustrated in Figure 19. Each sub-image in Figure 19 displays a hand pose overlaid with red landmarks and skeletal lines representing detected keypoints. The corresponding command is displayed in red text.

The gesture recognition was performed in real time with minimal latency (typically under 200 ms), and the system maintained high reliability under standard lighting conditions. The experimental results demon–strate that the hand gesture control module can be effectively used for intuitive, non-contact control of the robotic arm.

The integration of computer vision techniques into robotic control opens up a more intuitive and engaging way for users, especially students, to interact with robotic systems. In this project, two vision-based interaction methods were implemented: face tracking and hand gesture recognition. These approaches not only serve practical functions in controlling the robot's position and gripper, but also provide students with valuable hands-on experience in interdisciplinary topics.

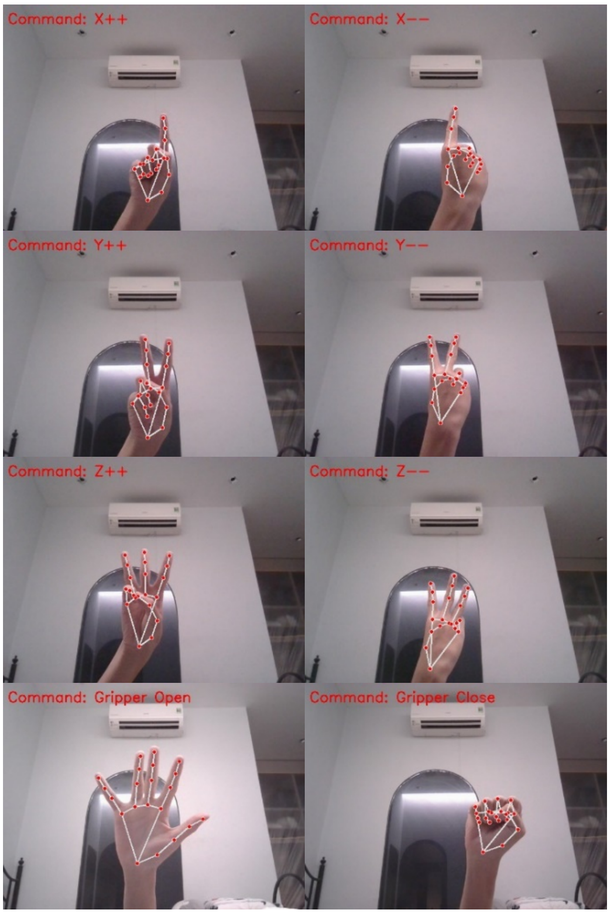


Figure 19. Hand gesture recognition results mapped to robot control commands.

Through the face tracking exercise, students are introduced to concepts such as object detection, coor–dinate transformation, and inverse kinematics. By ob–serving how the robot responds to their head movements in real time, they gain an understanding of how image-space data can be translated into physical-world robot coordinates, and how joint angles are calculated to reach those positions.

The hand gesture control activity reinforces under–standing of human-computer interaction, real-time land–mark detection, and state-based control logic. Students learn how distinct hand poses can be classified and mapped to robot commands, enhancing their knowledge of pattern recognition and control systems.

Overall, these exercises help students build a foundational understanding of AI, computer vision, and robotics, while fostering computational thinking, prob–lem-solving skills, and creativity. More importantly, they offer a tangible demonstration of how theoretical concepts in mathematics (e.g., trigonometry, coordinate geometry), programming, and control engineering come together in real-world robotic applications.

Figure 20 illustrates a model setup for simulating the operation of a production line, where the robot performs the tasks of picking up and sorting objects transported on the conveyor. An RGB camera is fixed above the conveyor to observe objects as they move through the work area. Images from the camera are processed by a trained YOLOv12 model, which is capable of accurately recognizing common objects in the classroom, such as ballpoint pens, markers, erasers, etc.

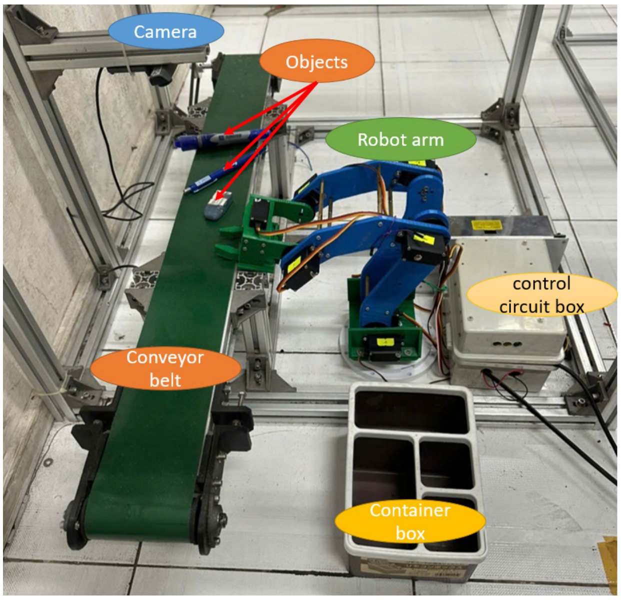


Figure 20. Experimental setup of the pick-and-place production line model with robotic arm, conveyor belt, vision system, and sorting containers.

After the object is detected, the system extracts the coordinates of the bounding box and converts them to actual coordinates on the conveyor plane. These coor–dinates are then calculated kinematically to convert them into rotation angles for each robot joint, enabling the gripper to move accurately to the object's location.

Finally, the robot uses a gripper to pick up the object and place it in pre-classified positions according to its type (for example, a pen in tray A, a marker in tray B, etc.). This process replicates an industrial automated product sorting system, demonstrating the effective combination of computer vision, artificial intelligence, and precision robotic control systems.

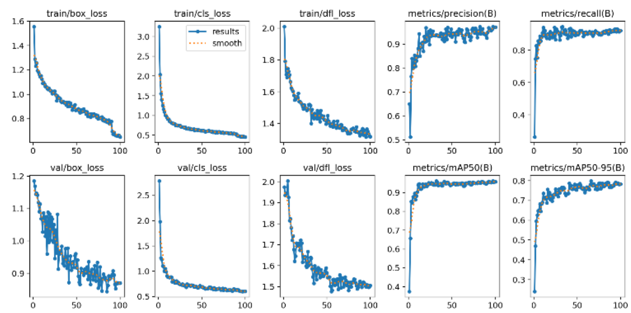


Figure 21. Training loss and evaluation metrics of the YOLOv12 model.

Figure 21 illustrates the training process of the YOLOv12 model on a dataset of common objects in a school environment, including pens, pencils, and erasers. The graphs showing the metrics such as train/loss, val/loss, precision, recall, and mean Average Precisionat IoU threshold of 0.5 (mAP@0.5) all show a steady improvement trend over each epoch. This indi–cates that the model learns effectively and has the ability to generalize well.

Figure 22. Confusion matrix on the validation dataset.

Figure 22 is the confusion matrix after training the model. Here, labels such as "eraser", "highlight", "mar–ker", "pen", and "background" all achieve high accu–racy. Some small confusion between objects of the same type (e.g., between a pen and a marker) is acceptable in a real environment.

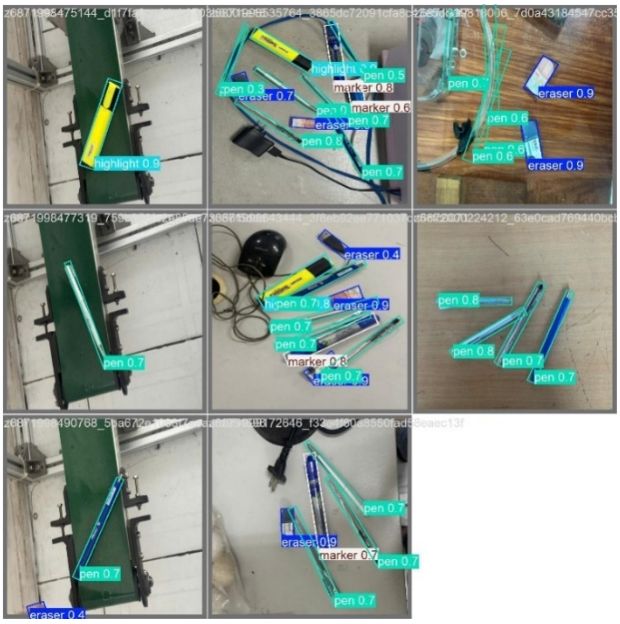


Figure 23. Object detection results in real classroom scenarios.

Figure 23 shows the actual detection results with objects placed on the conveyor belt. The bounding boxes and labels are assigned with clear confidence scores, showing that the system can handle multiple objects simultaneously and assist the robot in locating and classifying the object to be picked.

This exercise provides students with an opportunity to understand and apply artificial intelligence in a real-world context, specifically in the field of robotic automation. By simulating a smart production line using a robotic arm and a conveyor belt, learners gain insight into how robots are used in industrial environments for tasks such as object detection, classification, and automated sorting.

Through the data collection and model training process, students become familiar with essential steps in developing an AI system. These include capturing and annotating images, organizing datasets, training object detection models (such as YOLOv12) using platforms like Roboflow and Google Colab, and validating model performance. During this process, students also learn how to interpret key evaluation metrics, such as preci–sion, recall, Mean Average Precision (mAP), and loss functions, which helps them develop critical skills in assessing AI effectiveness.

Beyond the software component, this activity emp–hasizes the integration of AI with hardware systems. Students learn how object detection outputs (e.g., bounding box coordinates) are translated into robot joint movements using inverse kinematics. The combination of computer vision, AI models, servo motors, a conve–yor belt, and gripper control demonstrates how multiple subsystems can be orchestrated to achieve intelligent automation.

By completing this exercise, students not only dee–pen their understanding of robotic systems but also develop interdisciplinary competencies that span AI, computer vision, mechanical control, and real-time sys–tem integration, key areas in modern STEM and engineering education.

6. CONCLUSIONS

This paper presented the development of a low-cost, open-source robotic arm platform for STEM education, with a focus on integrating mechanical design, embed–ded control, and intelligent perception. The system comprises a 4-DOF robotic manipulator driven by servo motors, an ESP32-based embedded controller for real-time operation, and dual control interfaces supporting both local (PyQt5) and remote (IoT-based) access. Additionally, computer vision modules powered by YOLO object detection and MediaPipe-based face and hand tracking were integrated to enable interactive, AI-driven manipulation tasks.

Experimental demonstrations validated the system's functionality across multiple educational scenarios. Stu–dents were able to control the robot via facial tracking and hand gestures, detect and pick objects using real-time vision, and interact with the platform through both GUI and cloud-based interfaces. The results confirm that the platform effectively supports interdisciplinary learning, covering topics in mechanics, electronics, programming, AI, and human–robot interaction. More–over, the platform's modular and extensible design en–sures scalability and adaptability to a range of educa–tional levels and curricula.

For future work, several enhancements are envisi–oned. These include implementing inverse kinematics for automated trajectory planning, adding depth sensing for 3D perception, and extending the AI modules with learning-based grasp planning or voice interaction. Further classroom evaluations across different age groups and education settings will also be conducted to quantitatively assess the platform's impact on learning outcomes and engagement. Ultimately, the proposed system contributes toward bridging the gap between traditional educational robotics and the intelligent, connected systems demanded by Industry 4.0.

ACKNOWLEDGMENT

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

REFERENCES

[1] Y. Xie, M. Fang, K. Shauman, "STEM Education," Annual Review of Sociology, Vol. 41, pp. 331-357, 2015.

[2] S. Aboudahr, S. Aslam, L.U. Hua, A. Misron, S. Raime, Y.B. Wah, "Mapping the global landscape of STEM education: a bibliometric analysis using Scopus database," International Journal of Evaluation and Research in Education (IJERE), Vol. 13, No. 6, pp. 4225-4236, 2024.

[3] M.E. Andrews, A. Boklage, "Examining Students' (Inter-) Disciplinary Practice Beliefs in STEM Makerspaces," Education Sciences, Vol. 14, No. 12, 2024. https://doi.org/10.3390/educsci14121382

[4] F. Ouyang, W. Xu, W. (2024). The effects of edu–cational robotics in STEM education: A multilevel meta-analysis. International Journal of STEM Education, Vol. 11, No. 1, pp. 1-18, 2024. https://doi.org/10.1186/s40594-024-00469-4

[5] J. Varela-Aldás, Palacios-Navarro, "A ROS-Based Open Tool for Controlling an Educational Mobile Robot," International Journal of Online and Bio–medical Engineering (iJOE), Vol. 20, No. 01, pp. 23–39, 2024. https://doi.org/10.3991/ijoe.v20i01.46573

[6] Mateus-Coelho, N., Cruz-Cunha, M.M., Ávila, P.S.: Application of the Industry 4.0 Technologies to Mobile Learning and Health Education Apps, FME Transactions, Vol. 49, pp. 876-885, 2021.

[7] Francalanza, E., Borg, J., Rauch, E., Putnik, G.D., Alves, C., Lundgren, M., Amza, C.:Specifications for a Digital Training Toolbox for Industry 4.0, FME Transactions, Vol. 49, pp. 886-893, 2021.

[8] F. Mondada, E. Bonnet, S. Davrajh, W. Johal, and R. Stopforth, "R2T2. International Journal of Advanced Robotic Systems," International Journal of Advanced Robotic Systems, Vol. 13, No. 5, pp.1-13, 2016. https://doi.org/10.1177/1729881416658165

[9] C. Zeng, H. Zhou, W. Ye, X. Gu, X. (2021). IArm: Design an Educational Robotic Arm Kit for Inspiring Students' Computational Thinking. Sensors, Vol. 22, No. 8, 2021. https://doi.org/10.3390/s22082957

[10] S. Bargagna, E. Castro, F. Cecchi, F. et al., "Educational Robotics in Down Syndrome: A Feasibility Study," Tech Know Learn, Vol. 24, pp. 315–323, 2019. https://doi.org/10.1007/s10758-018-9366-z

[11] D.T. Tran, D.H. Truong, H.S. Le et al., "Mobile robot: automatic speech recognition application for automation and STEM education," Soft Comput, Vol. 27, No. 10789–10805, 2023. https://doi.org/10.1007/s00500-023-07824-7

[12] Haruna, A. I., Sankar, R., Samaila, A. (2023). Design and development of an instructional mobile robot for effective learning of material handling in mechanical workshops in universities. Materials Today: Proceedings. https://doi.org/10.1016/j.matpr.2023.05.617

[13] S. Pedre, M. Nitsche, F. Pessagc, J. Caccavelli, P. De Cristóforis, "Design of a Multi-purpose Low-Cost Mobile Robot for Research and Education," In: Advances in Autonomous Robotics Systems. TAROS 2014. Lecture Notes in Computer Science, Vol. 8717. Springer, Cham. 2014. https://doi.org/10.1007/978-3-319-10401-0_17

[14] J. Vega, V. Pérez, "G-ARM: An open-source and low-cost robotic arm integrated with ROS2 for educational purposes," Multimed Tools Appl, 2025. https://doi.org/10.1007/s11042-025-20748-8

[15] M. Eaton and M. H. Tanveer, "The Development and Implementation of a Cost-Effective Educati–onal Robotic Arm Using ROS-MoveIT," 2024 IEEE Integrated STEM Education Conference (ISEC), Princeton, NJ, USA, 2024, pp. 1-4, doi: 10.1109/ISEC61299.2024.10664704.

[16] L. ČehovinZajc, A. Rezelj, D. Skočaj, "Low-Cost Open-Source Robotic Platform for Education," in IEEE Transactions on Learning Technologies, vol. 16, no. 1, pp. 18-25, 1 Feb. 2023, doi:10.1109/TLT.2022.3224227.

[17] I. Chavdarov, K. Yovchev, B. Naydenov and V. Hrosinkov, "3D Printed DELTA Robot for Educa–tional Purposes," 2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2024, pp. 1-6, doi: 10.23919/SoftCOM62040.2024. 10721779.

[18] J. Vega, J.M. Cañas, "PiBot: An Open Low-Cost Robotic Platform with Camera for STEM Edu–cation," Electronics, Vol. 7, No. 12, 430, 2018. https://doi.org/10.3390/electronics7120430

[19] X. Zheng, "Robot Translation Based on Computer Vision for Cultural Psychology of English Culture Education," International Journal of Humanoid Robotics, Vol. 20, No. 02n03, 2023.

[20] A., H., Antonio, J., Alberto, C., F., L. "Educational Mechatronics Training System Based on Computer Vision for Mobile Robots," Sustainability, Vol. 15, No. 2, 1386, 2022. https://doi.org/10.3390/su15021386

[21] Z. Mamatnabiyev, C. Chronis, I. Varlamis, Y. Himeur, M. Zhaparov, "A Holistic Approach to Use Educational Robots for Supporting Computer Science Courses. Computers," Vol. 13, No. 4, 102, 2024. https://doi.org/10.3390/computers13040102

[22] S. Manzoor, R. Ul Islam, A. Khalid, A. Samad, J. Iqbal, J. “An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation,” Robotics and Computer-Integrated Manufacturing, Vol. 30, No. 3, pp. 351-362, 2014. https://doi.org/10.1016/j.rcim.2013.11.003.

[23] MediaPipe. MediaPipe Face Detection. https://mediapipe.readthedocs.io/en/latest/solutions/face_detection.html

[24] Phuong, L.H., Cong, V.D.: Control the Robot Arm through Vision-Based Human Hand Tracking, FME Transactions, Vol. 52, 37-44, 2024.

[25] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., Grundmann, M. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. ArXiv. https://arxiv.org/abs/2006.10214

[26] Phuong, L.H., Trung, P.X., Quyen, P.T., Mai, T.T.T.: Development of an Intelligence Vision for a Robot System to Pick and Place Objects, FME Transactions, Vol. 53, pp. 233-242, 2025.

[27] Roboflow. The Fastest Way to Label Computer Vision Datasets. https://roboflow.com/annotate

NOMENCLATURE

θ_1	Base rotation joint angle (rotation about vertical Z-axis)
θ_2	Vertical-plane joint angle (shoulder)
θ_3	Vertical-plane joint angle (elbow)
θ_4	Gripper angle
(X, Y, Z)	End-effector position in robot Cartesian coordinates
(u, v)	Pixel coordinates of the bounding-box centroid in the image plane
s	Face bounding-box area
W_f, H_f	Width and height of the video frame
k_x, k_y, k_z	Scaling factors in mapping from (u, v, s) to (X, Y, Z)
l_i	Link length(s) in the kinematic model
d_0	Vertical offset from the base

Acronyms and abbreviations

AC	Alternating Current
AI	Artificial Intelligence
C1-C5	Step size levels
DC	Direct Current
DC-DC	DC-to-DC Converter
DOF	Degree(s) of Freedom
GPIO	General-Purpose Input/Output
GUI	Graphical User Interface
IoT	Internet of Things
IoU	Intersection over Union
K-12	Kindergarten to 12th grade
MQTT	Message Queuing Telemetry Transport
PWM	Pulse Width Modulation
RGB	Red–Green–Blue
ROS	Robot Operating System
STEM	Science, Technology, Engineering, and Mathematics
USB	Universal Serial Bus
YOLO	You Only Look Once
YOLOv12	YOLO version 12
mAP	mean Average Precision
mAP@0.5	mean Average Precision at IoU=0.5

РАЗВОЈ ОБРАЗОВНЕ РОБОТСКЕ ПЛАТФОРМЕ СА ДВОСТРУКИМ КОНТРОЛНИМ ИНТЕРФЕЈСИМА И ИНТЕЛИГЕНТНОМ ИНТЕРАКЦИЈОМ ЗАСНОВАНОМ НА КОМПЈУТЕРСКОМ ВИДУ

Д.Ф. Тин, В.Т.Н. Хан, Н.Н. Тин, Н.В.Д. Хуј, Т.К. Туан, Н.Н. Дуј, Ф.Т. Дат, В.К. Хиеу, В.Д. Конг

Овај рад представља развој јефтине образовне ро–ботске платформе са омогућеном вештачком инте–лигенцијом дизајниране да побољша практично СТЕМ учење. Систем се састоји од роботске руке са 4 степена слободе (DOF) конструисане од 3Д штампаних компоненти, контролисане помоћу кола заснованог на ESP32 и интегрисане са локалним и веб-базираним интерфејсима за флексибилан рад. Развијена су два контролна интерфејса: локални десктоп ГУИ изграђен са PiKt5 и удаљени веб-базирани интерфејс који користи МКТТ протокол. Оба интерфејса омогућавају корисницима да ручно контролишу углове зглобова робота, прилагођавају величину корака покрета, враћају се на подразу–меване позиције и прате стања зглобова у реалном времену. Ови интерфејси пружају интуитивну инте–ракцију, омогућавајући ученицима да разумеју контролу покрета у роботици. Да би се интегрисала вештачка интелигенција и компјутерски вид, имп–лементирана су три модула: праћење лица, контрола покрета рукама и детекција објеката. Праћење лица преводи положај и величину лица у 3Д координате за кретање робота у реалном времену користећи инверзну кинематику. Препознавање покрета ру–кама користи МедиаПипе за тумачење поза прстију и извршавање одговарајућих радњи робота. Модул за детекцију објеката користи YOLOv12 модел обучен на објектима у учионици (оловке, гумице, маркери) за обављање аутономних задатака бирања и постављања на симулираном транспортном систему. Експери–ментални резултати потврђују ефикасност система у праћењу у реалном времену, интерпретацији покрета и манипулацији објектима. Графикони углова згло–бова и координата радног простора у реалном времену илуструју одзивно понашање система. Ова интегрисана платформа омогућава студентима да истражују вештачку инте–лигенцију, визију и робо–тику у јединственом окружењу. Кроз интерактивне

вежбе стичу прак–тично искуство у програмирању, развоју АИ модела, дизајну корисничког интерфејса и роботској контроли, повезујући теоријско знање и примене у стварном свету.