Robotic Arm using Computer Vision

Karma Kelzang Eudon[1], Dawa Zangpo[2], Bishal Rai[3], Karma Wangchuk[4], Nima Zangmo[5], Karma Norbu[6]

Department of Instrumentation and Control Engineering, College of Science and Technology, Royal University of Bhutan
E-mail: karmakelzang.cst@rub.edu.bt[1], zangpodawa05@gmail.com[2] , bishalrai230798@gmail.com[3], karma101wangchuk@gmail.com[4] , nimazangmo031@gmail.com[5], karmanorbu.cst@rub.edu.bt[6]

Abstract
This project outlines the creation of a computer vision-based robotic arm through a systematic approach. The design and implementation of the robotic arm integrates a computer vision system to automate complex tasks that further enhance the precision and adaptability of the system. Moreover, the integration of the camera makes the system more advanced, as image processing and image segmentation allow the machine to perform under different circumstances and adapt to new scenarios. Such systems can be used in various fields such as healthcare and gas station.
Key Words: *Computer Vision, Robotic arm, Image processing, Image segmentation*

1. INTRODUCTION

1.1. Background

The introduction of robotics and automation is often perceived by some as being limited to technology research and development (Covaciu et al., 2020). However, the primary purpose of robotics and automation lies in enhancing and supporting human activities. The integration of robotics into science and technology represents one of humanity's most remarkable achievements. Among these advancements, robotic arms which are programmable machines designed to execute tasks with precision and efficiency have garnered significant attention for their versatility and potential. These robotic manipulators are widely utilized across diverse domains, ranging from office settings to industrial applications, where they automate complex processes and contribute to improved productivity.

1.2. Literature Review

Robotic arms are used in manufacturing, healthcare, aerospace, and exploration industries. Jain et al. (2023) assert that articulated robotic arm for garbage disposal in hospital environments. On the other hand, the robotic arm can be used to pick, place, and grade fruits based on their efficiency (Dairath et al., 2023).
The design of robotic arms involves choices regarding kinematics, end-effector design, and materials. Dragos Cracium et al. (2022) constructed a robotic arm controlled using human hand movements, where they used the concept of exoskeleton sleeve design. The mechanical design used is based on the robot manipulator that has functions similar to those of the human arm (Elfasakhany, 2011).

1.3. Problem Statement

Mohammed et al. (2022) state that traditional robotic manipulation systems are not always reliable with novel objects in cluttered environments. Therefore, most robots need human intervention and are not autonomous. Moreover, integrating robotic arms with advanced computer vision techniques presents its challenges.

1.4. Aim and Objectives

Aim
To develop a robot arm system enhanced by computer vision (OpenCV) to sort, pick and place objects based on colour.
Objectives
1. Develop a computer vision system that can identify objects based on colour (red, green and blue).
2. Design and build a robotic arm.
3. Enable interaction between vision and manipulation.
4. Implement real-time control algorithms

2. METHODOLOGY

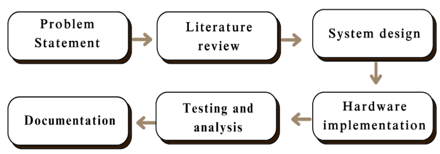


Fig.1: *Methodology*

The development of a robotic arm necessitates a systematic and methodical approach, commencing with the identification of the problem statement. This is followed by an extensive literature review to gather insights into the state-of-the-art automation and robotics systems currently in use. Subsequently, the design phase is initiated, where the robotic arm is meticulously conceptualized and constructed using precise measurements, pre-calculated torque values, and an optimized kinematic configuration. Furthermore, the system is integrated with compatible sensors and actuators to ensure the creation of a robust and functional robotic arm. During the hardware implementation phase, attention is directed toward assembling hardware components and establishing an efficient wiring system. Testing plays a pivotal role in the development process, focusing on evaluating the accuracy, speed, and reliability of the robotic arm through rigorous performance assessments and validation of the wiring setup. Finally, thorough documentation is essential for future reference, providing detailed records of the development process and serving as a critical resource for ensuring the safety and long-term functionality of the system.

3. FLOWCHART

1. Start: It is the initialization of the system.
2. Home position: The position of robotic arms at the beginning, acts as a reference point for all other positions and actions.
3. Color detection: the system examines coloured objects. This decision node branches into two possible paths:
   - Not detected: Loops back to the home position and continues scanning if colour is not detected.
   - Color detected: If colour is detected, the system proceeds to the colour identification process.
4. Color Identification: Upon detecting the colour, the system will check if the object is coloured red, green, or blue. For all the colours mentioned, there is a different path. These will perform the extraction of the coordinates for each coloured object:
   Green
   - No: It checks with other colours — red and blue.
   - Yes: If a green colour is detected, it extracts the x-y coordinate and moves on to calculate the joint angles.

   Red
   - No: It loops back to scan for colour.
   - Yes: If red colour is detected, it extracts the x-y coordinate and moves on to calculate the joint angles.

   Blue
   - No: It loops back to scan for colour.
   - Yes: If a blue colour is detected, it extracts the x-y coordinate and moves on to calculate the joint angles.
5. Calculating the Angles of Joints: The joint angles required for the robotic arm to move to pick and place each of the stated colours are calculated using the obtained coordinates through inverse kinematics.
6. Servo Motor Rotation: The servo motors are rotated using the calculated joint angles allowing them to reach specific positions and perform the physical task of picking and placing of objects.
7. Pick and Place: The robot picks up the specified object and places it in the designated location, and then the arm rotates back to the home position.

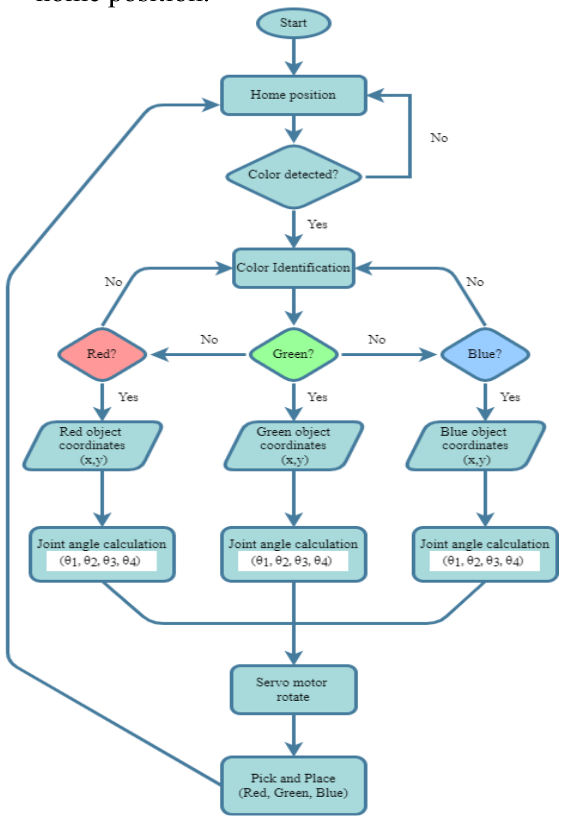


Fig.2: *Flowchart*

4. SYSTEM ARCHITECTURE

The schematic diagram of the robotic arm system begins with the switched-mode power supplies (SMPS), which provide the necessary power to the motors via the motor driver. The motor driver plays a critical role in supplying power to the motors while regulating their speed. This driver is connected to five motors that serve as the primary actuators for the joints of the robotic arm. An ESP32 microcontroller is utilized to transmit control signals to the motors, thereby enabling the precise movement of the robotic arm. The ESP32 microcontroller interfaces with a laptop, which facilitates programming and data exchange. For vision capabilities, the system incorporates a webcam. The laptop is responsible for performing image processing, while the ESP32 handles inverse kinematics computations and angle manipulation, ensuring accurate and coordinated motion of the robotic arm.

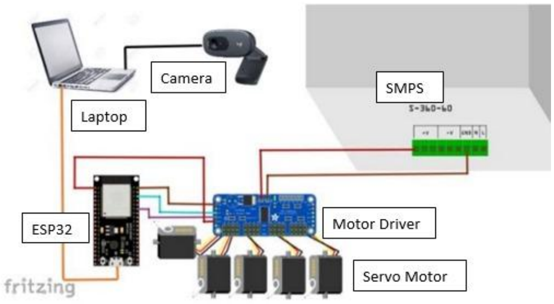


Fig.3: *System architecture*

5. TORQUE CALCULATION

The torque requirements for each joint of the robotic arm are calculated based on the formula $T=F*L$. The mass of a link is obtained from the estimated mass of PLA required to print it, and the length of links is taken as per the model design. The prototype set-up considered for calculations is fully stretched. To provide an analogy for this consideration, it is very difficult to hold a bucket of water when our arm is fully stretched. However, it's not true if our arm is bent to a certain angle. Hence, consideration of the robotic arm as fully stretched helps in determining the worst-case scenario for the torque required for each joint. The calculation for required torque is carried out from the manipulator towards the base

Table 1: *Torque calculation*

Sl. no.	Components	Weight (kg)	Length (m)	Torque (kg-cm)
1	Payload	0.15	-	-
2	SG90	0.009	-	1.2
3	MG996	0.06	-	13
4	MG995	0.04	-	10
5	Gripper	0.0042	0.05	0.76
6	Link 1	0.0428	0.08	2.51
7	Link 2	0.11	0.11	6.23
8	Link 3	0.142	0.13	12.77
9	Base	0.144	0.035	13

6. HARDWARE AND SOFTWARE COMPONENT

6.1. Software Components

a. Solidworks

SolidWorks is software that is widely used by engineers and encompasses both CAD (Computer-Aided Design) and CAE (Computer-Aided Engineering). The software provides a platform for modelling, simulation, visualization, and design, facilitating designers and engineers in developing and analyzing design factors quickly and precisely.

b. PrusaSlicer

PrusaSlicer is open-source slicing software that is used for the Prusa 3D printer. The software is flexible and offers various options for different file formats such as STL, 3MF, OBJ, AMF and others. Moreover, it operates for other 3D printers as well. PrusaSlicer takes in 3D models, sets up paths, and provides G-code for the 3D printer to print the object. The software was used to 3D print the parts of the robotic arm.

c. Visual Studio Code

Visual Studio Code, also known as VS code is an open-source code editor established by Microsoft for various operating systems such as Windows, Linux and MacOS. VS code is widely used by developers as it provides numerous features and supports various programming languages such as Python, Java, C++ and many more.

d. PlatformIO

PlatformIO is a cross-platform integrated development environment (IDE) that comes as an extension in the Visual Studio Code. It is user-friendly and provides many powerful features for embedded software development. It supports over 20 frameworks and more than 1500 microcontroller boards with over 13000 libraries. For this project, PlatformIO was used to program ESP32 microcontroller

e. OpenCV

OpenCV, also called Open-Source Vision Library is the free software for computer vision and machine learning (ML). The software has many functionalities including computer vision, image processing, ML and more.

f. MATLAB

MATLAB (Matrix Laboratory) is a high-level programming language widely used for numerical computation and other engineering applications. Some of its key features include numerical computation, interactive programming environment, graphics, and toolboxes. It also includes Simulink, which will be used for system simulation.

g. Fritzing

Fritzing is an open-source software that enables users to assemble various electronic components and build a PCB (Printed Circuit Board) outline for their projects. Fritzing offers a wide range of features such as circuit design, breadboard, and schematic views, among others.

6.2. Hardware Components

Table 2: *Materials for the robotic arm*

Components	Quantity
SG 40 servo motor	1
Servo motor MG995	2
Servo motor MG996R	2
Webcam	1
Jumper Wire	APR
Servo Motor Driver (PCA9685)	1
ESP32 microcontroller	1
SMPS	1

7. COMPUTER VISION

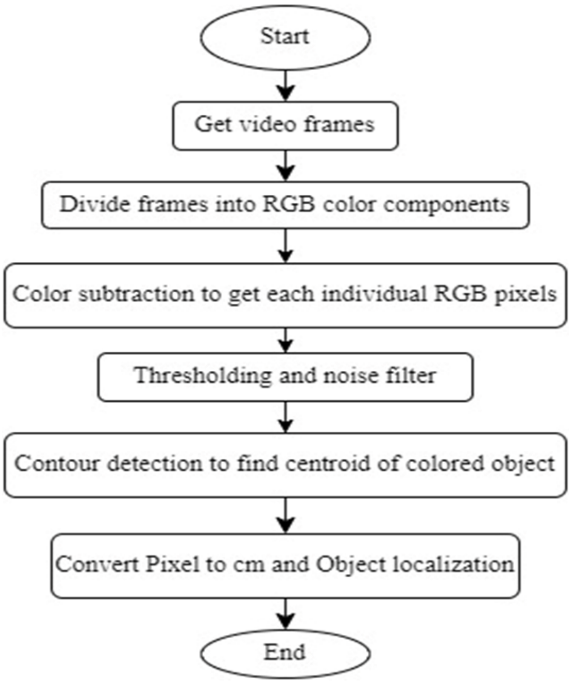


Fig.4: *Open CV flowchart*

The flow chart provided describes the computer vision system for this project, which detects coloured objects and identifies their locations with respect to the base of the robot so that the robot can sort, pick, and place these coloured objects.

8. KINEMATICS

8.1. DH Convention

For the diagram of the serial manipulator, the Denavit-Hartenberg convention is used to assign the frames to each joint according to the design and rotation of servo motors

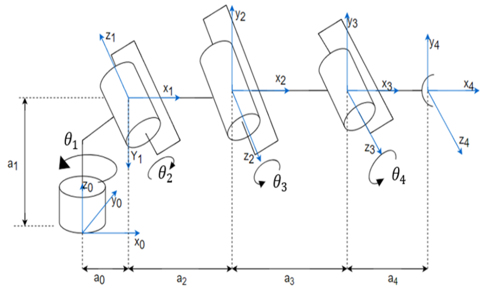


Fig.5: *DOF serial manipulator*

8.2. Inverse Kinematics

Inverse kinematics calculates the required joint angles for the given end effector position, fulfilling the requirements for this project. While, the geometric method is used to calculate the inverse kinematic solutions, decomposing the geometry of the robot arm from 3D to 2D.

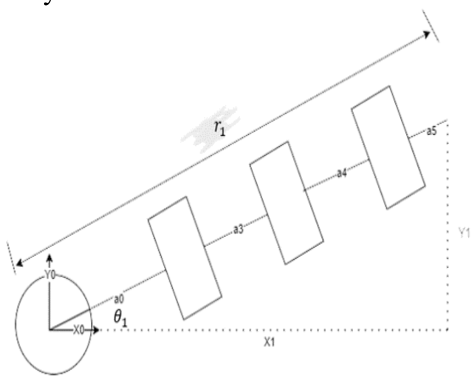


Fig.6: *Top view*

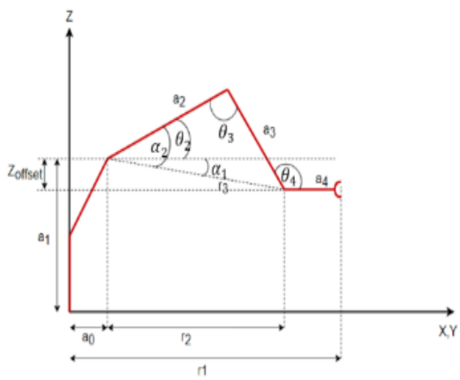


Fig.7: *Side view*

Table 3: *DH parameter table*

n	θ(deg)	α(deg)	R (cm)	d (cm)
1	θ_1	-90	a0	a1
2	θ_2	180	a2	0
3	θ_3	0	a3	0
4	θ_4	0	a4	0

$$\theta_1 = \left(\frac{Y}{X}\right)$$

$$r_1 = \sqrt{(X)^2 + (Y)^2}$$

$$r_2 = r_1 - a_4 - a_0$$

$$Z_{offset} = a_1 - Z$$

$$r_3 = \sqrt{r_2^2 + Z_{offset}^2}$$

$$\alpha_1 = \left(\frac{r_2}{r_3}\right)$$

$$\alpha_2 = \left(\frac{r_3^2 + a_2^2 - a_3^2}{2r_3a_2}\right)$$

$$\theta_2 = \alpha_1 + \alpha_2$$

$$\theta_3 = \left(\frac{a_3^2 + a_2^2 - r_3^2}{2a_3a_2}\right)$$

$$\theta_4 = 180 - \{[180 - (\alpha_2 + \theta_3)] - \alpha_1\}$$

8.3. Customization of GUI in MATLAB

The main purpose of designing a user-friendly interface in MATLAB is to visualize the orientation and position of the robot model when subjected to different kinematic parameters before actual prototype deployment.

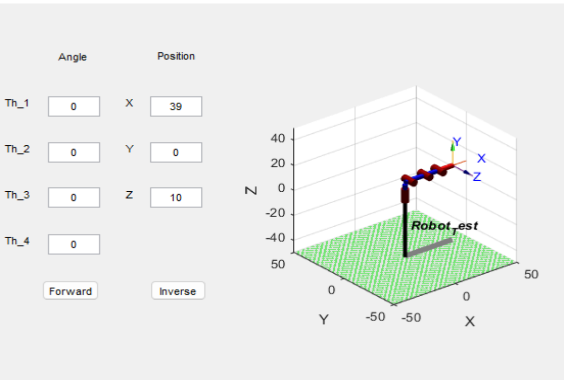


Fig.8: *Result for given parameter*

9. DESIGN AND PROTOTYPE

In this design, servo motors are mounted externally. This implementation has resolved many problems previously encountered in other designs. Since the motors are mounted outside the arm, the design is sleek and functional; a good proportion of the bulkiness that the robot would have had was reduced. With this configuration, serving or replacing the motors is easier. Furthermore, the good weight distribution has improved control, thus enhancing the stability and performance of the entire arm system.

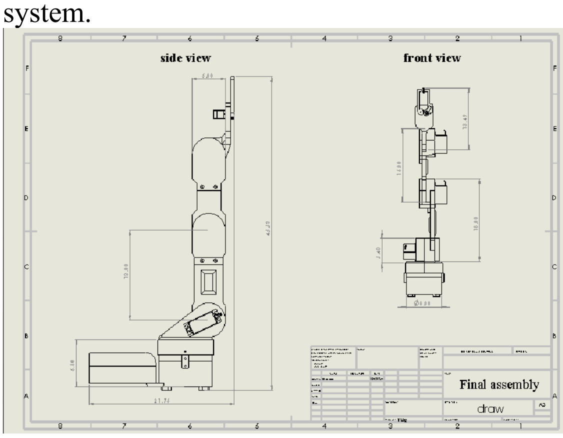


Fig.9: *Design of robotic arm*

10. TESTING AND ANALYSIS

10.1. Accuracy Testing

In robotics, maintaining accuracy is crucial for the efficient execution of tasks. Accuracy is defined as the difference between the desired position and the obtained position. For our prototype, we made it to repeatedly pick the object and place it in a defined position. The prototype was subjected to different delays in milliseconds, and errors in millimeters were obtained, summarized in the tables below.

Table 4: *Error in accuracy along x-axis*

X-dir.	*Error [mm], delay: 20ms*	*Error [mm], delay: 30ms*	*Error [mm], delay:40ms*
1	-7	-5	-4
2	-5	-2	-1
3	-7	-5	-2
4	-2	-4	-2
5	-4	-2	-1
6	-6	-5	-2
7	-9	-4	-1
8	-6	-5	-5
9	-9	-5	-4
10	-8	-5	-2

Table 5: *Error in accuracy along y-axis*

Y-dir.	*Error [mm], delay: 20ms*	*Error [mm], delay: 30ms*	*Error [mm], delay:40ms*
1	5	2	1
2	5	2	1
3	3	3	2
4	4	1	-1
5	5	1	-2
6	5	5	3
7	2	3	4
8	5	4	2
9	5	6	3
10	5	6	1

The accuracy is calculated by:

$$AP_p = \sqrt{\left(\left(\underline{x} - x_c\right)^2 + \left(\underline{y} - y_c\right)^2\right)}$$

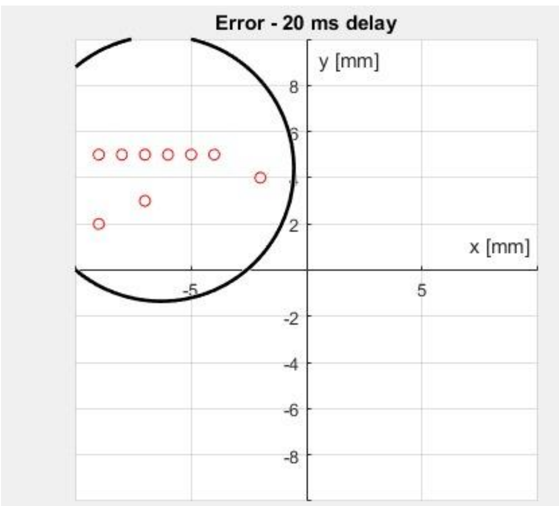
$$\underline{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \qquad \underline{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$$

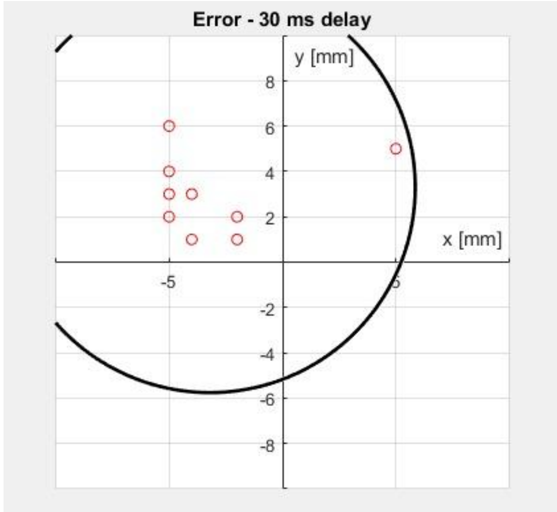
The results of the calculations are as follows:

Table 6: *Result of accuracy*

Delay (*ms*)	Accuracy (*mm*)
20	7.98
30	5.34
40	2.70

The results obtained indicate that the accuracy can be significantly improved by increasing the delay.





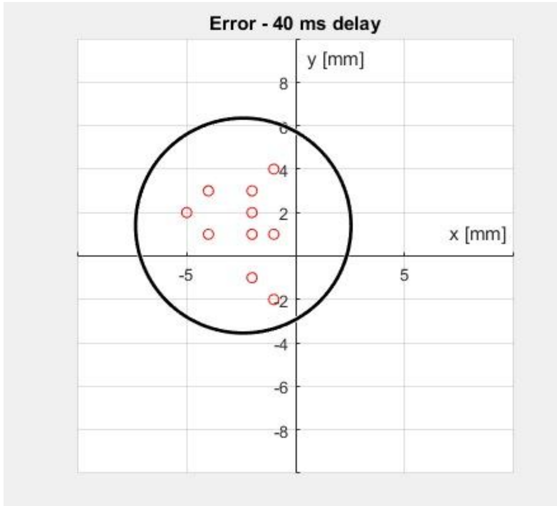


Fig.10: *The placement of object to the target position under various delays*

In the figure above, the accuracy of the test can be seen. The red dots indicate the coordinates where the robotic arm placed the object during the accuracy testing. The commanded position is the origin (0,0). The plots are made using the MATLAB software

11. RESULT AND RECOMMENDATION

11.1. Final Prototype

The final prototype design of the robotic arm is a fusion of both hardware and software components. The arm links are joined by the servo motors and are controlled by the ESP32 microcontroller. The power supply is given by the SMPS and the type of end effector used is a gripper. For movement planning, inverse kinematics is used.

11.2. Challenges Faced

Some of the challenges faced while working on the project include:

1. Initial lack of knowledge in robotics: The project began with no prior knowledge on robotics, requiring a considerable amount of time and effort to learn the basics and principles.

Fig.11: *Final prototype*

2. Camera calibration: Calibrating the camera with a robot presented a significant technical challenge.
3. Maintain mechanical precision: To maintain precise mechanical components, a well-designed and executed manufacturing process was essential. The linkages and joints were all manufactured and assembled correctly.
4. Component failure: The prominent problem occurred due to the component failure. Components like motors were prone to failure due to various reasons.

11.3. Future Scope

The type systems and innovations can potentially be used in many various fields such as:

1. Healthcare: The robotic arm can be utilized in hospitals and pharmacies for sorting medications, ensuring accurate and efficient dispensation. This application enhances precision in medication handling, minimizes errors, and streamlines the workflow in healthcare settings.
2. Gas Stations: The robotic arm system can be integrated into gas stations to automate the dispensing of gasoline or other fuels, replacing human operation. This integration enhances efficiency, speed, and safety, offering a more reliable and streamlined fueling process.
3. Agriculture: The robotic arm can be employed in agricultural technology, particularly for tasks such as harvesting ripened fruits. Equipped with advanced object detection capabilities and an integrated camera, the robotic arm can be programmed to identify and selectively pick only ripe strawberries, enhancing efficiency and reducing labour requirements in farming operations.
4. Education: The robotic arm serves as an exceptional tool for STEM Education providing young learners with hands-on practical experience in robotics, programming, and automation. It facilitates a deeper understanding of engineering, computer science, and technology while fostering creativity and cultivating a spirit of innovation among aspiring students.

11.4. Recommendation

This 4 DOF robotic arm can be further upgraded to elevate the capability of the robotic arm, making it more versatile, efficient, and intelligent in performing a wide range of tasks. It can be achieved by implementing the following recommendations:

1. Attaching the Camera to the Robotic Arm: There are some advantages of having the camera rigidly mounted on the manipulator over having it mounted rigidly in parallel to the workspace. In these, the camera can move following the arm, creating dynamic, real-time perception from various angles and perspectives.
2. Wireless communication: Implementing wireless communication methods over serial communication can significantly improve the system's flexibility and range. Wireless communications such as Wi-Fi or other form such as Bluetooth, in these functions of the robotic arm can increase efficiency in the process of deploying the robotic arm in several settings without the constraints posed by wired connections.
3. Machine Learning or AI for Better Object Identification: Implementing the advanced techniques for detecting objects: ML, or AI, among others, can further increase the accuracy of the robotic arm, and reliability.

12. CONCLUSION

In conclusion, this project, which involved the development of a robotic arm utilizing computer vision, demonstrated the system's capability to perform complex tasks such as object picking, placement, and sorting. The team successfully designed and implemented a fully functional system leveraging OpenCV for object identification and kinematics for precise movement. The project encompassed various stages including design, hardware implementation, and rigorous testing.

Throughout the developmental process, several challenges were encountered. These included maintaining mechanical precision, enhancing the system's overall accuracy, and addressing hardware issues such as component failures. However, with appropriate guidance from subject experts, these challenges were effectively resolved, culminating in an efficient robotic arm system.

REFERENCES

Arents, J., & Greitans, M. (2022). Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Applied Science*, *12*(2), 2. https://doi.org/10.3390/app12020937

Clothier, K. E., & Shang, Y. (2010). A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation. *Journal of Robotics*,*2010*,1–10. https://doi.org/10.1155/2010/984823

Elfasakhany, A., Yanez, E., Baylon, K., & Salgado, R. M. (2011). Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom. *Modern Mechanical Engineering*,*01*(02),47–55. https://doi.org/10.4236/mme.2011.12007

Dairath, M. H., Akram, M. W., Mehmood, M. A., Sarwar, H. U., Akram, M. Z., Omar, M. M., & Faheem, M. (2023). Computer vision-based prototype robotic picking cum grading system for fruits. *Smart Agricultural Technology*, *4*, 100210. https://doi.org/10.1016/j.atech.2023.100210

Hsiao, J., Kumar, S., Chou, C., & Kam, T. (2020). Shape design optimization of a robot arm using a Surrogate-Based evolutionary approach. *Applied Sciences*, *10*(7), 2223.https://doi.org/10.3390/app10072223

Intisar, M., Khan, M. M., Islam, M. R., & Masud, M. (2021). Computer Vision based robotic arm controlled using interactive GUI. *Intelligent Automation and Soft Computing*,*27*(2),533–550. https://doi.org/10.32604/iasc.2021.015482

Jain, S., Monish, M., Gupta, N., Raj, S.k., & Subramanian, K. (2023). Articulated robot arm for garbage disposal in hospital environment. *ITM Web of Conferences*, *56*, 01002.https://doi.org/10.1051/itmconf/20235601002

Pérez, R., Rubert, S. C. G., & Zotovic, R. (2018). A study on Robot arm Machining: advance and future challenges. In *Annals of DAAAM for & proceedings of the International DAAAM Symposium* (pp. 0931–0940). https://doi.org/10.2507/29th.daaam.proceedings.134

Mohammed, M. Q., Kwek, L. C., Chua, S. C., Al-Dhaqm, A., Nahavandi, S., Eisa, T. a. E., Miskon, M. F., Al-Mhiqani, M. N., Ali, A., Abaker, M., & Alandoli, E. A. (2022). Review of Learning-Based Robotic Manipulation in Cluttered Environments. *Sensors*, *22*(20), 7938. https://doi.org/10.3390/s22207938

Moran, M. E. (2007). Evolution of robotic arms. *Journal of Robotic Surgery*, *1*(2), 103–111. https://doi.org/10.1007/s11701-006-0002-x

Yusoff, M. a. K., Samin, R. E., & Ibrahim, B. (2012). Wireless Mobile Robotic Arm. *Procedia Engineering*, *41*, 1072–1078. https://doi.org/10.1016/j.proeng.2012.07.285