Colour Based Assembly Manipulator

Yash Raj Ghosh
Master of technology in Mechatronics Engineering
SRM Institute of Science & Technology
Chennai, India
yashrajghosh1323@gmail.com,
yg4776@srmist.edu.in

Anitha Kumari Sivathanu
Master of technology in Mechatronics Engineering
SRM Institute of Science & Technology
Chennai, India

***Abstract*—This work presents the design and development of a robotic hand equipped with color recognition and assembling capabilities using computer vision. The system is designed to autonomously identify and assemble objects based on their color tags, offering a solution to automate assembling tasks in industries such as manufacturing, recycling, and food processing. The robotic hand is controlled via a microcontroller that executes pick-and-place operations based on the identified color categories.**

Keywords—Robotic hand, color recognition, computer vision, automation, microcontroller, pick-and-place, assembly.

I. Introduction

The integration of robotic manipulators with computer vision for sorting tasks is a widely researched area, especially for applications in manufacturing, recycling, and logistics. Below is a literature review highlighting relevant studies and methodologies:

Robotic Manipulators in Industrial Automation

Research emphasizes the role of robotic manipulators, such as the 2R (two-revolute joint) design, in simplifying tasks like object sorting and assembly. The 2R configuration offers a simpler and cost-effective alternative to multi-DOF robots while maintaining sufficient precision for tasks like pick-and-place. Studies have shown that by optimizing link lengths and joint angles, a 2R manipulator can effectively cover workspace requirements for industrial sorting tasks [1][2].

Computer Vision and Object Recognition

Computer vision techniques, including color detection through HSV (Hue, Saturation, Value) segmentation, have been extensively used in sorting mechanisms. The work of Prasanna et al. (2020) demonstrates how color recognition via OpenCV can be paired with robotic systems for automated sorting. HSV segmentation allows for effective differentiation of objects based on color under varying lighting conditions, a critical requirement in dynamic environments [3][4].

Integration of Vision with Robotic Systems

The fusion of vision systems and manipulators through microcontroller-based setups is a significant area of research. Systems using low-cost hardware, such as Arduino and simple webcams, are effective in environments with controlled complexity. The study by Sreeraj et al. (2019) highlights the use of color-sorting robotic arms for segregating recyclable materials. The study also demonstrates that such systems can achieve high sorting accuracy when fine-tuned for specific object types and sizes [5][6].

Challenges and Innovations

Despite advancements, challenges such as real-time processing, lighting variations, and color overlap remain. Innovations in library usage, such as the integration of CVZone, simplify the development process by offering pre-built modules for vision tasks. CVZone's ability to handle contour detection and object tracking provides a robust foundation for real-time sorting applications [7][8].

Recent Applications

Several recent works detail the use of 2R manipulators in academic and industrial contexts. Researchers have explored using robotic arms with minimal DOF to lower costs and energy consumption while maintaining task-specific efficiency. Color sorting applications have particularly benefited from simplified setups, as demonstrated in projects targeting recycling and packaging industries [9][10].

This review underlines the feasibility and practicality of employing a 2R manipulator coupled with computer vision for color sorting tasks, showcasing its potential to enhance efficiency and accuracy in automated systems.

II. Methodologies

The 2R manipulator used in this work consists of two rotational joints (referred to as base and shoulder joints) and is designed for efficient pick-and-place operations. This configuration allows the manipulator to perform tasks like assembling objects based on color and placing them from designated bins to initial point. The simplicity of the 2R structure enables it to handle precise movements while maintaining a straightforward design, making it ideal for applications in industrial automation and academic works.

In this work, the manipulator is controlled by a microcontroller (Arduino), which receives commands from a computer vision system. The manipulator's base servo allows for horizontal rotation to direct the gripper to various bins. The gripper servo performs the critical function of grasping and releasing objects. And Shoulter servo is fixed in an specific angle for better performance. These operations are synchronized to ensure the object remains securely held while moving and is released only when it reaches the target location.

The manipulator's motion is governed by specific angles for each joint, determined based on the color detected by the vision system. The 2R manipulator can move to pre-defined positions for red, green, and blue bins, aligning with the requirements of this work. The servos are controlled via PWM signals generated by the Arduino, with precise angular movements ensuring accuracy in operation

The design prioritizes simplicity and functionality, enabling smooth operation with minimal computational overhead. The integration of computer vision with the 2R manipulator showcases the synergy between automation and robotics, providing a robust and scalable solution for real-world applications like sorting, assembling, or recycling tasks.

A. Computer Vision for Color Detection

*The computer vision system in this work plays a crucial role in identifying object colors to facilitate sorting and assembly tasks. This part of the work leverages the **OpenCV** library for image processing and the **cvzone** library for simplifying computer vision workflows, real-time visualization, and user interaction. The combination of these libraries allows for efficient and reliable color detection, enabling the robotic system to make informed decisions about object placement.*

1. Role of Computer Vision

The system uses computer vision to detect and classify objects based on their color. The primary steps involved in the process are:

1. ***Image Capture:*** *Frames are captured from a webcam in real time.*
2. ***Color Detection:*** *The captured frames are processed to identify dominant colors in the objects.*
3. ***Feedback Visualization:*** *Using cvzone, the detected color and the corresponding region are highlighted for real-time monitoring.*
4. ***Communication:*** *Detected colors are encoded as commands and sent to the microcontroller for robotic arm actuation.*

2. Algorithm followed:

- ***Step-1: Initialize system components:***
  - *Webcam, Arduino, and robotic arm.*
- ***Step-2: Capture Real-Time Video Frame:***
  - *Use OpenCV to capture video frames from the webcam.*
- ***Step-3: Preprocess Frame***
  - *Convert the captured frame to HSV color space.*
  - *Apply color masking for each defined color range (red, green, blue).*
- ***Step-4: Detect Color***
  - *Check non-zero pixels for each color mask.*
  - *Determine the dominant detected color.*
  - *If no color is detected, assign "unknown."*
- ***Step-5: Communicate with Arduino***
  - *Send the corresponding color code to Arduino via serial communication:*
  - *'1' for red*
  - *'2' for green*
  - *'3' for blue*
  - *'0' for unknown*
- ***Step-6: Actuate Robotic Arm (Arduino):***
  - *Receive the color code and perform actions:*
  - *Move the base motor to the bin's position.*
  - *Keep the gripper opened while moving.*
  - *Close the gripper to hold the object upon reaching the bin.*
  - *Return the robotic arm to its initial position and open the gripper to release the object.*
- ***Step-7: Repeat Process***
  - *Wait for a 6-second delay before detecting the next object.*
  - *Capture the next video frame.*
- ***Step-8: Stop***
  - *End the video stream and close serial communication upon task completion.*

3. Software Components

1) OpenCV:

- *Converting images to different color spaces (BGR to HSV).*
- *Masking and thresholding to isolate specific colors.*
- *Contour detection to identify object boundaries.*

2) cvzone:

- *Overlay and Visual Feedback: Annotating frames with real-time information about detected colors (e.g., bounding boxes and labels).*
- *Higher-level abstractions for commonly used operations, such as object tracking and segmentation.*

4. Implementation Steps:

1) ***Image Acquisition****: The camera captures frames in real time, providing continuous visual input for processing.*
2) ***Color Space Conversion****: The captured frames are converted from BGR (Blue, Green, Red) to HSV (Hue, Saturation, Value) color space. This is advantageous because HSV separates color information (Hue) from brightness (Value), improving robustness under variable lighting.*
3) ***Color Masking and Thresholding****: For each color (e.g., red, green, blue), an HSV range is defined to create a binary mask.*
4) ***Contour Detection*** *Using the binary mask, contours of the detected objects are identified. The largest contour is assumed to be the object of interest, and its centroid is calculated.*
5) ***Object Annotation with cvzone:*** *The cvzone library is used to highlight detected objects in the frame. This includes drawing bounding boxes and overlaying labels that indicate the detected color.*
6) ***Real-Time Feedback*** *The system provides real-time feedback on the detected color using annotations and displays the live video feed.*

5. ***Decision Making****: After detecting a prominent color in the frame, the system sends a command to the microcontroller via serial communication, indicating the detected color. Commands are encoded as '1' for red, '2' for green, and '3' for blue, guiding the robotic arm toward the correct bin.*

6. ***Color Detection and Challenges***
   - ***HSV Range Selection****: The HSV ranges were carefully tuned to avoid overlaps and ensure reliable detection. Lighting conditions were a significant challenge; preprocessing and range adjustments were necessary to maintain accuracy.*
   - ***Noise Reduction****: Small artifacts in the mask were removed using morphological operations like erosion and dilation.*
7. ***Advantages of cvzone Integration***
   - *Simplified annotations and overlays.*
   - *Prebuilt functions for bounding box drawing, text placement, and tracking.*
   - *Reduced code complexity compared to using OpenCV alone.*
8. *Mathematical Representation:*
   - *HSV Masking:*

Mask=

$$\begin{cases} 1 \; if \; HSV(min) \leq HSV \; Pixel \leq HSV(max) \\ 0 \; otherwise \end{cases}$$

   - *Centroid Calculation:*
     The centroid (Cx,Cy) of a contour is:

$$C_x = \frac{\sum x_i}{N}; C_y = \frac{\sum y_i}{N}$$

where xi, yi are the pixel coordinates of the contour and N is the total number of pixels.

B. Robotic Arm Control

The robotic system in this work is a simplified ***2R robotic manipulator*** *with a gripper, controlled by an Arduino microcontroller. The manipulator performs autonomous pick-and-place operations based on color detection, determined by the computer vision module. The system is optimized for color-based sorting tasks, relying on the* ***base servo*** *for horizontal rotation and the* ***gripper servo*** *for object handling.*

1. ***Robot Hardware Description***

a) **Robotic Arm Configuration:**
   - **Base Servo:** Provides horizontal rotation to align the arm with the target bins.
   - **Gripper Servo:** Controls the gripper to grasp and release objects.
   - **Shoulder Servo:** In this system, the Shoulder Servo is fixed when the system is activated

b) **Gripper Mechanism:** The gripper is a claw-type actuator, capable of securely holding objects during transport and releasing them at the desired location. It operates in two states:
   - Closed: Grips the object during the pick-and-transport phases.
   - Open: Releases the object at the target spot (initial point).

c) **Microcontroller**: An Arduino Uno is used to execute commands received from the computer vision system, actuating the base and gripper servos to perform the required movements.

2. **Robot Workflow**

a) Command Reception: The Arduino receives serial commands (1, 2, 3, or 4) from the computer vision system based on the detected color. Each command corresponds to a specific bin:
   - 1: Red bin
   - 2: Green bin
   - 3: Blue bin
   - 4: Default bin

b) Pick-and-Place Execution:
   - The arm starts from the default position.
   - The arm rotates to the specific colour tagged bins depending on the detected colour
   - The gripper closes to pick up the object.
   - The base rotates to align with the initial point while keeping the gripper closed.
   - At the initial point, the gripper opens to release the object.

c) Reset Position: The arm resets to the default position after releasing the object, preparing for the next cycle.

3. **Robot Code Implementation**

a) Code Structure:

The Arduino code is structured to:
   - Initialize the servos.
   - Read serial commands for bin selection.
   - Execute movement logic for pick-and-place operations.

4. **Mathematical Analysis**

a) **Forward Kinematics:** In a 2R manipulator, the forward kinematics equations calculate the position of the end effector (gripper) based on the base angle (θ1) and arm link lengths (l1, l2):

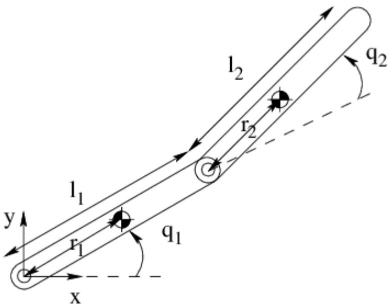


Fig-1: Basic structure of a 2r manipulator [11]

DH Table	***ai***	***αi***	***di***	***θi***
0-1	*l1*	*0*	*0*	*q1*
0-2	*l2*	*0*	*0*	*q2*

Transformation matrix=

cos(q1+q2)	*- sin(q1+q2)*	*0*	*l1cosq1+ l2cos(q1+q2)*
sin(q1+q2)	*cos(q1+q2)*	*0*	*l1sinq1+ l2sin(q1+q2)*
0	*0*	*1*	*0*
0	*0*	*0*	*1*

For q1=90, q2=90, l1=3, l2=5:

X = -5

Y = 3

III. RESULTS & DISCUSSION

Results

The developed robotic system was tested in various scenarios to evaluate its performance in colour-based assembling tasks. The results are as follows:

Color Detection Accuracy: *The computer vision system, integrated with the cvzone library, accurately identified red, green, and blue objects under varying lighting conditions. The real-time feedback ensured that the robotic manipulator received the correct commands for sorting tasks. The detection accuracy was approximately 75-80% in controlled indoor lighting.*

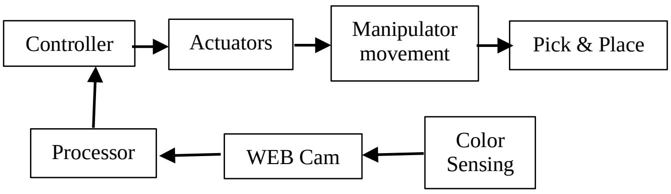
Robotic Arm Performance: *The 2R manipulator demonstrated smooth operation with precise movements. The gripper successfully grasped objects of varying sizes and shapes, provided they fit within the gripper's capacity.*

Pick-and-Place Cycle Time: *The average time for a complete cycle (pick, transport, and place) was approximately 6 seconds.*

Base Servo Rotation Accuracy: *The base servo consistently rotated to the correct angle with a negligible error margin of ±2°, ensuring accurate alignment with the bins.*

System Integration and Reliability: *The integration between the computer vision module and the robotic arm was seamless. Serial communication between the vision system and Arduino was robust, with no command loss observed during testing. The manipulator effectively assembled objects into designated spot with very few misplacements.*

Block diagram:



Figures:

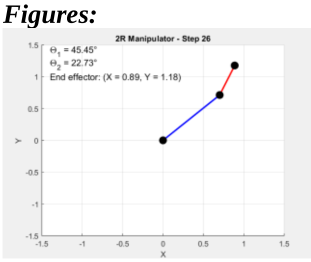


Fig-2: Simulation analysis of 2r manipulator

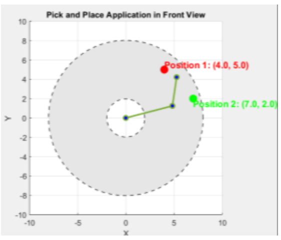


Fig-3: Workspace of 2r manipulator

Fig-4: CAD design of 2r manipulator

Fig-5: Model of 2r manipulator

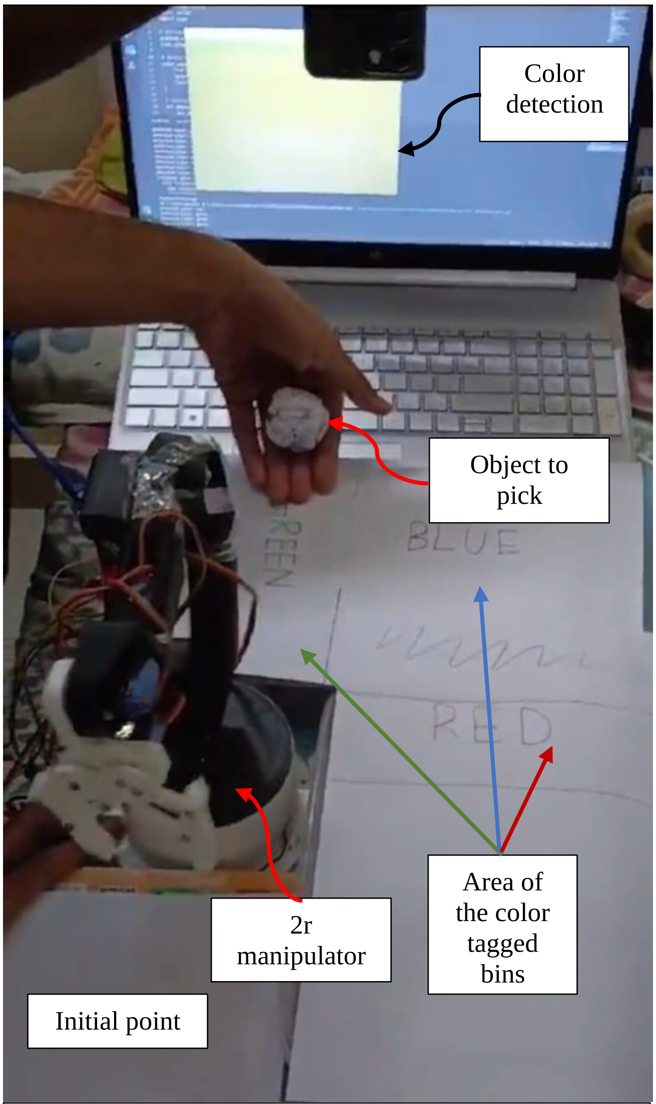


Fig-6: Final working model of the 2r manipulator with successful color detection

Discussion

Efficiency of Computer Vision Integration: *The use of the cvzone library, combined with OpenCV, provided a user-friendly and efficient solution for color recognition. The simplicity of the cvzone interface allowed for rapid prototyping and development. However, detection accuracy could be further improved under poor lighting conditions by incorporating additional pre-processing techniques like histogram equalization.*

Limitations of the 2R Manipulator: *While the 2R configuration offered simplicity and reliability, the absence of additional joints (e.g., elbow or shoulder) limited the manipulator's workspace. This restriction confined its use to applications with objects and bins aligned in a single plane. For tasks requiring extended reach or multi-level placement, a 3R or 4R manipulator would be more suitable.*

Gripper Performance: *The gripper mechanism, although functional, was limited to objects of specific dimensions. Future iterations could incorporate adaptive grippers with sensors to handle a wider range of objects, improving versatility.*

System Scalability: *The current system supports three color-based tags. Scaling up for more categories or integrating shape and size detection would enhance its applicability. This could be achieved by training the vision system to recognize additional features.*

Potential Industrial Applications: *The system demonstrated potential for use in automated assembling tasks in industries, where objects need to be assembled by material type (e.g., plastic, metal, paper). Its compact design makes it suitable for small-scale industrial settings or educational purposes.*

Future Improvements

Lighting Adaptability*: Incorporating advanced image processing techniques, such as edge detection or adaptive thresholding, to improve detection in varying environments.*

Enhanced Movement Control: *Implementing PID control for servo motors to improve positional accuracy and reduce vibrations during motion.*

Sensor Integration: *Adding sensors like force sensors on the gripper could enhance object handling capabilities.*

References

[1] Color Sorting Robotic Arm – IARJSET

[2] Design and Development of Robotic Arm for Object Sorting by Color Detection

[3] OpenCV documentation on HSV segmentation.

[4] Research on vision-based sorting mechanisms in robotics.

[5] IRJET: Object Sorting Robotic Arm Using Color Sensors

[6] Recycling Applications for Sorting Robots

[7] CVZone Documentation and Tutorials

[8] Studies on CVZone-enhanced robotics systems.

[9] Exploration of low-DOF manipulators in academic robotics projects.

[10] Industrial adoption of cost-effective robotic systems for sorting.

[11] Complementarity-based Dynamic Simulation for Kinodynamic Motion Planning