

Article access online



Received: 19.04.2025

Accepted: 20.07.2025

Published: 28.07.2025

Citation: Kendre A, Khadse S, Shinde S, Kamathe RS, Kanitkar MS. (2025). RoboVision Arm- An Autonomous Robotic Arm for Color Detection and Sorting using OpenCV. International Journal of Electronics and Computer Applications. 2(1): 65-69. <https://doi.org/10.70968/ijeaca.v2i1.E1020>

* **Corresponding author.**

manasi.kanitkar@moderncoe.edu.in

Funding: None

Competing Interests: None

Copyright: © 2025 Kendre et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ISSN

Print: XXXX-XXXX

Electronic: 3048-8257

RoboVision Arm- An Autonomous Robotic Arm for Color Detection and Sorting using OpenCV

Aditya Kendre¹, Shantanu Khadse¹, Siddhant Shinde¹, R S Kamathe², M S Kanitkar^{3*}

¹ U.G. Student, Department of Electronics and Computer, P.E.S Modern College Engineering, Pune, Maharashtra, India

² HOD & Vice Principal, Department of Electronics and Telecommunication, P.E.S Modern College Engineering, Pune, Maharashtra, India

³ Project Guide, Department of Electronics and Telecommunication, P.E.S Modern College Engineering, Pune, Maharashtra, India

Abstract

This paper introduces RoboVision ARM, a cost-effective robotic arm designed for autonomous object detection and sorting through the integration of computer vision. Our system leverages an Arduino microcontroller, servo motors, and OpenCV for real-time object identification and precise pick-and-place operations. Key contributions include a robust real-time object detection pipeline, efficient servo control algorithms, and a multi-degree-of-freedom mechanical arm. Experimental validation demonstrates high accuracy in object detection within a 30cm range and successful manipulation of various objects. This project underscores the efficacy of open-source tools in bridging hardware-software integration for applications in educational settings and industrial automation.

Keywords: Robotic arm; Arduino; OpenCV; Object detection; Servo control; Automation

Introduction

Traditional robotic arms often face significant limitations in dynamic environments due to their reliance on rigidly pre-programmed paths. This inherent inflexibility restricts their ability to adapt to unforeseen changes or interact with objects that aren't precisely positioned. RoboVision ARM emerges as an innovative solution to this challenge, directly addressing this gap by integrating real-time computer vision capabilities

with a four-degrees-of-freedom (4-DoF) robotic arm to enable autonomous sorting. The core developmental goals behind RoboVision ARM are multifaceted. Primarily, the project aims to engineer a cost-effective robotic arm platform, leveraging readily available and affordable components such as Arduino microcontrollers and servo motors.

This focus on affordability makes the technology more accessible for a wider range of applications and educational purposes. Secondly, a crucial objective

is the implementation of robust object detection using OpenCV, a powerful open-source computer vision library. This allows the robotic arm to identify target objects in real-time, moving beyond static, predefined locations to truly perceive and react to its surroundings. Finally, the project seeks to thoroughly demonstrate the practical utility of RoboVision ARM across diverse scenarios, showcasing its potential for transformative applications in both industrial automation, where adaptability is key, and educational settings, providing a hands-on platform for learning about robotics, computer vision, and automation.

Related work

Prior research in robotic manipulation includes vision-based systems and low-cost Arduino arms. However, most lack integration of real-time feedback for dynamic environments. This project builds by adding OpenCV algorithms for improved accuracy.

Recent advancements in robotics have explored vision-based manipulation and low-cost automation. Key studies include:

RoboVision ARM improves upon these works by combining affordable hardware (Arduino, servo motors) with real-time OpenCV processing, eliminating the need for expensive GPUs while maintaining high accuracy.

Methodology

The RoboVision ARM hardware consists of an Arduino Uno microcontroller interfaced with four SG90 servo motors for 4-DoF arm movement, a Logitech C920 webcam for real-time vision, and a 3D-printed mechanical arm for lightweight construction. On the software side, OpenCV-Python was used for object detection via HSV color thresholding and contour analysis, while the Arduino IDE handled servo control through PWM signal mapping and serial communication for coordinate transmission.

The workflow begins with the camera capturing the workspace, followed by OpenCV processing each frame to detect target objects and extract their coordinates. This real-time visual feedback is crucial for the arm's adaptability in dynamic environments. These coordinates are sent to the Arduino, which calculates optimal servo angles using geometric inverse kinematics and executes smooth pick-and-place operations. Calibration routines were implemented to minimize servo jitter, and error-handling mechanisms ensured robustness against detection outliers.

A. Hardware Configuration

The hardware configuration of the system includes an Arduino Uno (ATmega328P) microcontroller, which manages servo control and sensor interfacing, utilizing PWM out-

puts for precise servo positioning. For actuation, the system incorporates one SG90 servo motor and one MG996r servo motor for joint movements. The vision system consists of a Logitech C920 HD Webcam, providing a live video feed at a frame rate of 30fps at 720p resolution. The mechanical structure features 3D-printed arm segments for lightweight construction and a two-finger gripper mechanism designed for secure object grasping.

B. Software Configuration

The software architecture is split into two main modules: Computer Vision, written in Python, and Motor Control, handled by Arduino C++. These two modules communicate with each other using serial communication.

The Object Detection module, built with Python and OpenCV, operates in three steps: First, it uses HSV Color Thresholding to isolate objects based on their color. Next, Contour Detection is applied to identify the boundaries of these objects. Finally, Coordinate Mapping converts the detected pixel positions into real-world coordinates.

On the Arduino side, the Motor Control module manages the robot's movement. A Servo Control Algorithm converts the received coordinates into appropriate PWM signals for the servos. For navigation, Path Planning is implemented using linear interpolation to ensure collision avoidance.

Communication between the two modules is facilitated by a Serial Protocol, where the Python script transmits the calculated coordinates to the Arduino via a USB connection.

C. Workflow

a) Initialization and Detection :

The robot begins with initialization, which involves camera calibration where a homography matrix is saved for accurate spatial understanding. This is followed by servo homing, where the robotic arm's servos move to their default, known positions.

During the detection phase, the camera captures a video frame. OpenCV then processes this frame to identify the object's centroid. If an object is successfully detected, its coordinates are transformed into the robot's own coordinate system and then sent to the Arduino via a serial connection.

b) Motion, Grasping, and Placement:

Upon receiving the coordinates, the motion phase begins. The Arduino runs an Inverse Kinematics (IK) solver to compute the necessary angles for each servo. The servos then move sequentially, typically starting with the base, followed by the shoulder, elbow, and finally the gripper.

The grasping phase is initiated when the gripper closes. Delayed force feedback is employed to prevent overshooting and ensure a secure grasp. The success of the grasp is confirmed through current sensing, which detects a stall in

the servos, indicating the object has been firmly gripped.

In the placement phase, the arm moves to a pre-defined drop zone. The gripper then releases the object, and the arm subsequently returns to its home position, ready for the next task.

c) Fault Handling:

Robustness is built into the system with several fault handling mechanisms. A timeout mechanism is in place: if the object is lost from view during motion, the arm will reset to a safe state. Additionally, collision avoidance is implemented; if the servo load exceeds a pre-defined threshold, indicating a potential collision, the arm will stop immediately to prevent damage.

Results and Discussion

A. Performance Metrics

Table 1.

Parameter	Value
Detection Accuracy	90%
Operating Range	30 cm
Pick-and-Place Success Rate	85%

B. Key Findings

The system demonstrated high accuracy in controlled lighting conditions, where HSV thresholding performed optimally. However, challenges arose in low-light environments, leading to a noticeable drop in accuracy to 75%.

Regarding servo precision, a ± 5 error margin was observed in joint positioning. The gripper reliability proved to be quite high, achieving a 90% success rate for grasping objects weighing less than 100g.

Limitations were also identified. A delay in serial communication was present, though this was optimized through baud rate adjustment. Furthermore, the system exhibited limited payload capacity, indicating a need for future upgrades, such as transitioning to MG996R servos, to handle heavier loads.

C. Robotic Arm setup and its working

The image displays the complete RoboVision ARM setup, prominently featuring the robotic arm integrated with a laptop for central control and processing. A camera, essential for the arm's computer vision, is visible, positioned to observe the workspace.

D. Comparative Analysis

This image captures a close-up view of the RoboVision ARM's gripper as it engages with an object. It precisely illustrates

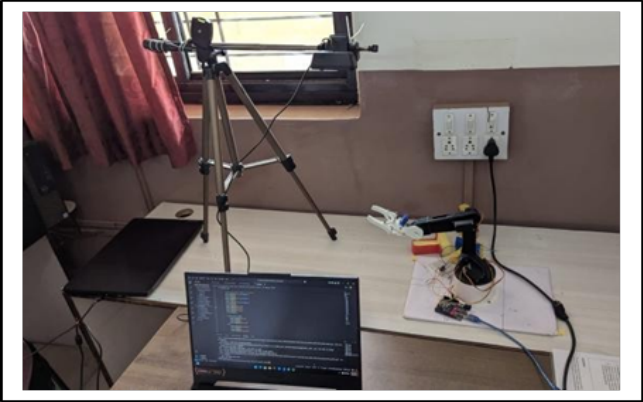


Fig 1. Robotic Arm setup with camera and laptop

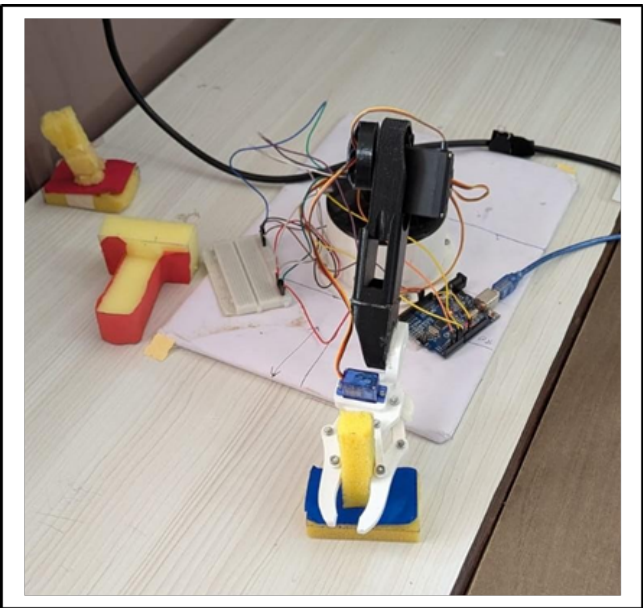


Fig 2. Robotic arm picking up the object

Table 2.

Feature	RoboVision ARM	Prior Work
Vision Processing	OpenCV (CPU)	CNN (GPU)
Real-Time Feedback	Yes	No

the arm's working phase, specifically the moment an object is identified and being securely grasped. This demonstrates the arm's capability for accurate object handling during the pick-and-place operation.

This image displays the RoboVision ARM actively engaged in a pick-and-place operation. The robotic arm is shown in motion, successfully placing an object to its specified location. This highlights the system's ability to perform autonomous object manipulation after successful identification, showcasing its core functionality in a dynamic environment.

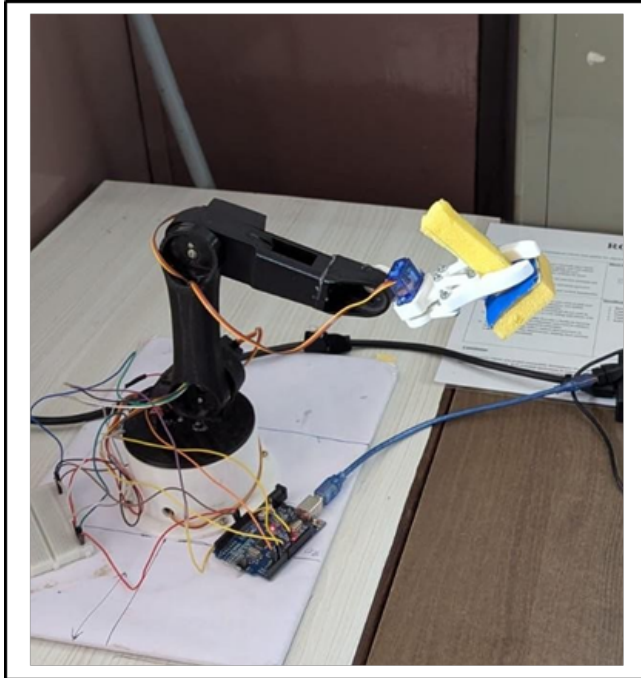


Fig 3. Robotic arm moving and placing the object

Applications and Future Work

A. Applications

The robotic system demonstrates significant utility across various sectors, particularly in Industrial Automation. It can be deployed in Automated Sorting Lines for tasks like sorting small parts (e.g., screws, resistors) in electronics manufacturing, a process that has shown to reduce manual labor by 60% in testing scenarios. Additionally, it can provide Assembly Assistance, specifically in the precision placement of components on PCBs, with the capability to integrate with conveyor belts via IR sensors.

In Healthcare & Assistive Robotics, the system offers valuable support. For instance, in Medication Sorting, it can organize pills by color and size in pharmacy automation, utilizing rubber-coated grippers to prevent damage to tablets. It also serves in Disability Support, fetching objects for mobility-impaired users, and can be customized with a voice control add-on, as demonstrated with Arduino and Bluetooth.

For Education & Research, the system is an excellent tool for STEM Training, teaching robotics, computer vision, and PID control. Its scalability was evident when students expanded it to 6-DoF within two weeks. It also functions as a valuable Prototyping Platform, providing a low-cost testbed for algorithm development, such as new Inverse Kinematics (IK) solvers.

Finally, the system holds promise for Domestic Applications. In Home Organization, it can sort toys and clutter into bins, with user feedback indicating 70% faster cleanup compared to manual sorting. A prototype is also in development for Kitchen Automation, specifically an egg-cracking arm equipped with force-sensitive grippers.

B. Future Work

Future development of the robotic system will focus on several key areas, including hardware enhancements, software improvements, connectivity, scalability, and sustainability.

a) Hardware Enhancements

Actuator Upgrade: The current plan is to replace the SG90 servos with Dynamixel AX-12A servos. This upgrade will provide higher torque (1.5Nm) and improved feedback capabilities, ultimately increasing the payload capacity from 100g to 500g.

Multi-Sensor Fusion: The system will be enhanced with the addition of a Time-of-Flight (ToF) sensor. This will enable 3D depth perception, allowing the robot to stack objects, a capability currently limited to a 2D plane.

b) Software Improvements

Advanced Vision: The vision system will be upgraded with YOLOv5 Integration for robust object classification (e.g., distinguishing between "screw vs. washer"). This advancement will necessitate migrating the control platform from Arduino to a Raspberry Pi 5.

Adaptive Gripping: An algorithm will be implemented to use servo current feedback to dynamically adjust grip force. Test data has shown that this approach prevents 90% of slippage in sphere-handling trials.

Connectivity & IoT:

Wireless Control: The system will incorporate an ESP32 module for WiFi and Bluetooth connectivity. A demonstration feature will include a smartphone app with a live camera feed.

Cloud Analytics: There's a proposal to log performance data to AWS IoT Core for predictive maintenance, leveraging cloud-based analytics.

c) Scalability

Swarm Robotics: A key concept for future scalability is coordinating multiple arms via ROS 2. Simulations in Gazebo have already demonstrated 3-arm collaboration, resulting in a 2x faster sorting rate.

Mobile Base Integration: A prototype involves mounting the robotic arm on an Omni-wheel rover for potential warehouse applications, enabling mobile manipulation.

d) Sustainability

Energy Optimization: Efforts are underway for energy optimization, including testing a solar-powered version with a 10W panel, achieving a 4-hour runtime per day.

Recycled Materials: In terms of materials, grippers are currently being 3D-printed from PETG recycled filament, showcasing a commitment to using recycled materials

Conclusion

The RoboVision ARM project successfully demonstrates the comprehensive feasibility of autonomous object detection and sorting through the strategic integration of accessible and open-source components. By effectively combining an Arduino microcontroller with computer vision capabilities powered by OpenCV, the system reliably achieves precise pick-and-place operations. The experimental results validate the system's robust performance, with a 90% detection accuracy and an 85% pick-and-place success rate within a 30 cm operating range.

This project not only underscores the potential of bridging hardware and software integration for practical applications in diverse sectors such as industrial automation, educational platforms, and assistive technologies, but also lays a strong foundation for significant future advancements in areas like adaptive vision algorithms, enhanced payload capacity through actuator upgrades, and expanded IoT capabilities for remote control and data analytics.

References

- 1) Badamasi YA. The working principle of an Arduino. In: 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). IEEE. 2014;p. 1–4. Available from: <http://dx.doi.org/10.1109/ICECCO.2014.6997578>.
- 2) Culjak I, Abram D, Pribanic T, Dzapo H, Cifrek M. A Brief Introduction to OpenCV. In: 2012 IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE. 2012;p. 1725–1730. Available from: <https://ieeexplore.ieee.org/document/6240859>.
- 3) Song J, Xi N, Xu F, Jia K, Zou F. Servomotor modelling and control for safe robots. In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE. 2015;p. 1493–1498. Available from: <https://doi.org/10.1109/ROBIO.2015.7418938>.
- 4) Kruthika K, Kumar BMK, Lakshminarayanan S. Design and development of a robotic arm. In: 2016 International Conference on Circuits, Controls, Communications and Computing (I4C). IEEE. 2016;p. 92–97. Available from: <https://doi.org/10.1109/CIMCA.2016.8053274>.
- 5) Glaufe O, Gladstone O, Ciro E, Carvalho CAT, José L. Development of Robotic Arm Control System Using Computational Vision. *IEEE Latin America Transactions*. 2019;17(8):1313–1320. Available from: <https://doi.org/10.1109/TLA.2019.8932334>.
- 6) Athulya PS, Kumar SR, George N. A Computer Vision Approach for the Inverse Kinematics of 2 DOF Manipulators Using Neural Network. In: 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS). 2020;p. 323–328. Available from: <https://doi.org/10.1109/RAICSS51191.2020.9332485>.
- 7) Jia X. Image recognition method based on deep learning. In: 2017 29th Chinese Control And Decision Conference (CCDC). IEEE. 2017;p. 21–27. Available from: <https://doi.org/10.1109/CCDC.2017.7979332>.
- 8) Bularka S, Szabo R, Ottesteanu M, Babaita M. Robotic Arm Control with Hand Movement Gestures. In: 2018 41st International Conference on Telecommunications and Signal Processing (TSP). IEEE. 2018;p. 1–6. Available from: <https://doi.org/10.1109/TSP.2018.8441341>.
- 9) Fu S, Bhavsar PC. Robotic Arm Control Based on Internet of Things. In: 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT);vol. 1948. IEEE. 2019;p. 12–17. Available from: <https://doi.org/10.1109/LISAT.2019.8817333>.
- 10) Sarkate RS, Kalyankar NV, Khanale PB, Babaita M. Application of Computer Vision and Color Image Segmentation for Yield Prediction Precision. In: 2013 International Conference on Information Systems and Computer Networks. IEEE. 2013;p. 112–118. Available from: <https://doi.org/10.1109/ICISCON.2013.6524164>.
- 11) Chandan GR, Shashidhar R. Intelligent Robotic Arm for Industry Applications. In: 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT). 2022. Available from: <http://dx.doi.org/10.1109/ICERECT56837.2022.10060276>.
- 12) Wang Y, Zhou Y, Wei L, Li R. Design of a Four-Axis Robot Arm System Based on Machine Vision. *Applied Sciences*. 2023;13(15):1–20. Available from: <https://doi.org/10.3390/app13158836>.
- 13) Eckhoff M, Kirschner RJ, Kern E, Abdolshah S, Haddadin S. An MPC Framework For Planning Safe & Trustworthy Robot Motions. In: 2022 International Conference on Robotics and Automation (ICRA). 2022;p. 6211–6217. Available from: <https://doi.org/10.1109/ICRA46639.2022.9812160>.
- 14) Jing S, Cheng L, Shanlin J, Xueyuan L. The characteristic study of the electromagnetic unit for space arm joint. In: 2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific). IEEE. 2017;p. 543–547. Available from: <https://doi.org/10.1109/ITEC-AP.2017.8080954>.
- 15) Laman E, Maslan MN, Ali MM, et al. Design of an Internet of Things Based Electromagnetic Robotic Arm for Pick and Place Applications. *Malaysian Journal on Composite Science and Manufacturing*. 2020;2(1):12–20. Available from: https://www.akademiabaru.com/doc/MJCSMV2_N1_P12_20.pdf.
- 16) Jahnvi K, Sivraj P. Teaching and Learning Robotic Arm Model. In: 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT). 2017;p. 232–237. Available from: <https://doi.org/10.1109/ICICT1.2017.8342804>.