Design and Development of Vision-Based Robotic Arm: An Integrated Approach for Object Detection and Autonomous Pick-and-Place Operations

Amudhan Rajarajan[1] Murasu Ramachandran[2] Sanmugapriyan S[3] Madesh M[4] Sugumar A[5]
[1,2,3,4,5]Department of Mechanical Engineering
[1,2,3,4,5]C.K. College of Engineering & Technology, Cuddaore, Tamil Nadu, India

Abstract — Vision-guided robotic systems have emerged as a key technology in Industry 4.0 by enabling robots to perceive and interact with dynamic environments. Conventional robotic manipulators typically rely on fixed coordinate programming, limiting their flexibility when object positions change. This paper presents the design and development of a low-cost vision-based robotic arm prototype capable of autonomous object detection and pick-and-place operations. The system integrates a 6-degree-of-freedom (6-DOF) robotic arm fabricated using Fused Deposition Modelling (FDM) 3D printing technology with computer vision techniques based on OpenCV and YOLOv8 object detection. A top-mounted camera captures real-time images of the workspace, and detected object positions are converted from image coordinates to real-world coordinates using homography-based calibration. Inverse kinematics algorithms are employed to compute robotic arm joint movements, while an Arduino UNO microcontroller controls servo actuation through a PCA9685 servo driver. Experimental evaluation demonstrated approximately 92% detection accuracy, coordinate mapping accuracy within ±0.5 cm, and successful autonomous pick-and-place operations under controlled laboratory conditions. The proposed system provides a cost-effective platform for intelligent automation, robotics education, and industrial material handling research.

Keywords: Vision-Based Robotics, Robotic Arm, YOLOv8, Object Detection, Computer Vision, Inverse Kinematics, Arduino, Homography Calibration, Pick-and-Place Automation, Industry 4.0

I. INTRODUCTION

A. *Background*

Precision nutrition is an emerging field that tailors dietary The rapid growth of Industry 4.0 has accelerated the adoption of automation technologies across manufacturing, logistics, and material handling sectors. Industrial robotic arms are extensively utilized for repetitive and precision-oriented tasks (J. Redmon et. al). However, most conventional robotic systems operate using predefined coordinates and lack the capability to adapt to dynamic environments where object locations vary continuously.

Recent advances in artificial intelligence, machine learning, and computer vision have enabled robots to acquire environmental awareness through visual perception (Z. Zhang et. al). Vision-guided robotic systems combine image processing with robotic control, allowing autonomous object identification, localization, and manipulation. Such systems significantly enhance flexibility and reduce dependence on manual programming (G. Bradski et. al).

B. *Problem Statement*

Traditional robotic arms require manual programming of object coordinates and workspace layouts. Any modification in object position or orientation necessitates reconfiguration of robot programs, resulting in reduced adaptability and increased setup time.

C. *Objectives of the Review*

The objectives of this study are:

- To design and fabricate a 6-DOF robotic arm using 3D printing technology.
- To develop a vision-based object detection system using YOLOv8 and OpenCV.
- To implement homography-based camera calibration for coordinate transformation.
- To formulate inverse kinematics for robotic arm positioning.
- To integrate Python-based vision processing with Arduino-based robotic control.
- To demonstrate autonomous pick-and-place operations in a laboratory environment.

II. METHODOLOGY

A. *System Architecture*

The proposed system consists of three primary subsystems:

- Vision Subsystem
- Computation Subsystem
- Actuation Subsystem

The vision subsystem utilizes a USB camera mounted above the workspace to capture real-time images. The computation subsystem executes object detection, coordinate mapping, and inverse kinematics calculations using Python. The actuation subsystem includes an Arduino UNO, PCA9685 servo controller, and six servo motors that drive the robotic arm (J. J. Craig et. al).

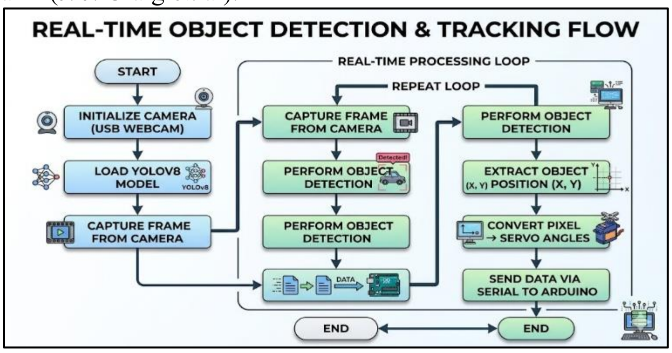


Fig. 1: Software Integration Flow

B. *Mechanical Design and Fabrication*

The robotic arm was designed using Autodesk Fusion 360 and fabricated through FDM 3D printing using PLA material (B. Sicilianoeat. al). The structure comprises:

- Base assembly
- Shoulder joint
- Elbow joint
- Wrist mechanism

– Gripper assembly

MG995 servo motors were employed for high-torque joints, while SG90 micro servos were used for wrist and gripper operations (M. W. Spong et. al).

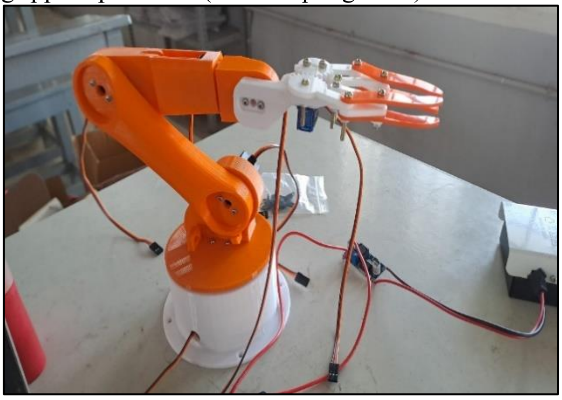


Fig. 2: Assembly of Robotic Arm

C. Vision-Based Object Detection

Real-time object detection was implemented using the YOLOv8 deep learning framework. A custom dataset containing more than 500 images of colored balls and cubes was collected and annotated using Labeling (P. Corke et. al).

The dataset consisted of:

– Red Ball
– Blue Ball
– Green Ball
– Yellow Ball
– Red Cube
– Blue Cube
– Green Cube
– Yellow Cube

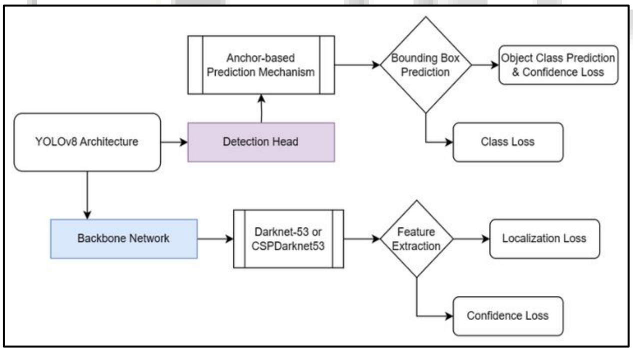


Fig. 3: YOLOv8 Working Flow diagram

The YOLOv8 model was trained using transfer learning and subsequently deployed for real-time inference.

III. Camera Calibration and Coordinate Transformation

A. Need for Calibration

Object locations detected by the camera are represented in pixel coordinates, whereas robotic manipulation requires physical coordinates (G. Jocher et. al). Therefore, a calibration process is necessary to establish a mapping between image space and workspace coordinates.

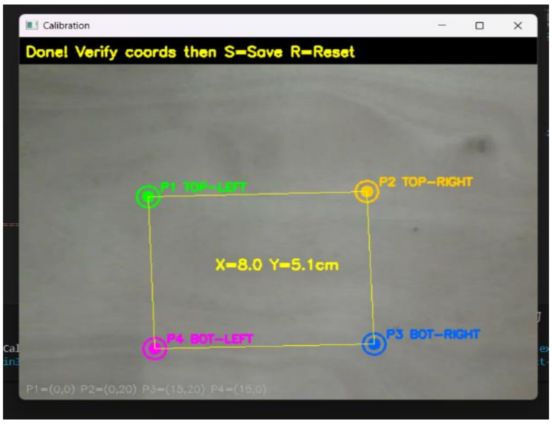


Fig. 4: Camera Calibration Sample images

B. Homography-Based Calibration.

A homography matrix was computed using known reference points in the workspace. The transformation converts image coordinates into real-world coordinates through projective mapping (Tzutalin et. al).

The calibration process included:

– Workspace definition
– Reference point identification
– Pixel coordinate extraction
– Homography matrix computation
– Storage of calibration parameters

This method provides an efficient solution for planar robotic workspaces.

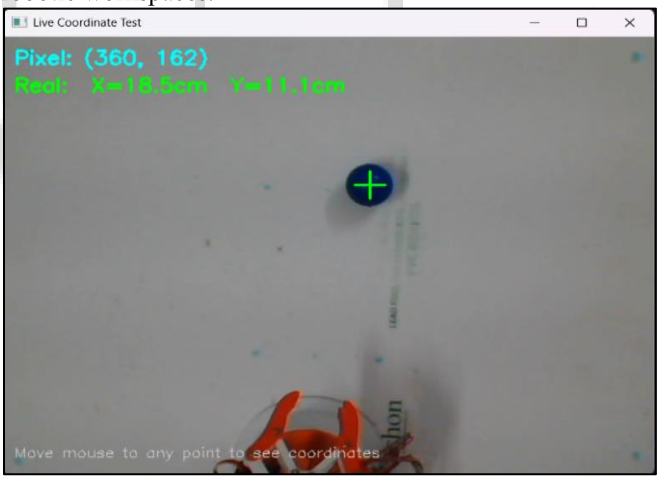


Fig. 5: Live Coordinate test (Pixel to Real world)

IV. Inverse Kinematics for Robotic Control

Inverse kinematics determines the joint configurations required to reach a desired object position (R. Szeliski et. al).

For object coordinates (X,Y), the base rotation angle was calculated using:

$\theta = atan^2(Y,X)$

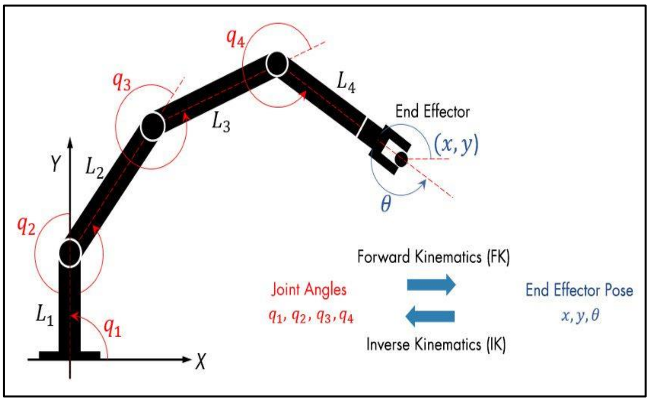


Fig. 6: Configuring the joint positions of a robotic arm using forward or inverse kinematics.

The calculated angle was mapped to corresponding servo pulse widths and transmitted to the Arduino controller. Safety constraints were incorporated to prevent mechanical over-rotation and collisions (D. Marr et. al).

The complete workflow consisted of:

Object Detection → Coordinate Transformation → Inverse Kinematics → Servo Command Generation → Robotic Actuation

V. STEP-BY-STEP OPERATION

The developed robotic arm system follows the sequence below during operation (A. Krizhevsky et. al).

Step 1 – System Initialization
- Power supply is switched ON
- Arduino initializes servo motors
- Python program starts camera and serial communication
- Robotic arm moves to HOME position

Step 2 – Image Capture
- USB camera continuously captures workspace images
- Frames are processed in real-time

Step 3 – Object Detection
- YOLOv8 model detects balls and cubes
- Bounding boxes and labels are generated
- Object centroid coordinates are extracted

Step 4 – Coordinate Conversion
- Pixel coordinates are converted into real-world coordinates using homography calibration
- Object position relative to robotic arm is calculated

Step 5 – Inverse Kinematics Calculation
- Base rotation angle is computed
- Servo pulse values are generated

Step 6 – Command Transmission
- Python sends commands such as:
- PICK
- PLACE
- HOME
- Commands are transmitted through serial communication

Step 7 – Pick Operation

The robotic arm performs:
- Base rotation
- Arm lowering
- Gripper closing
  - Object lifting

Step 8 – Place Operation
- The robotic arm moves the object to the designated drop location and releases it.

Step 9 – Return to Home Position
- Robotic arm returns to default HOME position
- System waits for next detection cycle

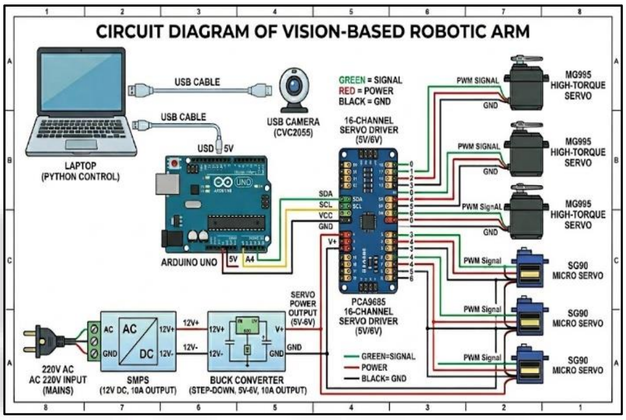
VI. RESULTS AND DISCUSSION

A. Object Detection Performance

The YOLOv8 model achieved robust real-time performance under laboratory conditions (Zhang et. al).

Performance Metrics:
- Detection Accuracy (mAP@0.5): ~92%
- Frame Rate: 25–30 FPS
- Multi-object Detection: Successful
- Detection Stability: High under controlled lighting

B. Coordinate Mapping Accuracy



The homography-based calibration achieved coordinate conversion accuracy within ±0.5 cm, enabling reliable robotic positioning.

C. Pick-and-Place Performance

The robotic arm successfully performed autonomous pick-and-place operations for all tested object categories (S. Russell et. al).

Observed Results:
- Pick Success Rate: ~90%
- Placement Accuracy: ±1 cm
- Full Cycle Time: 10–15 seconds
- Communication Reliability: No data loss observed

The integration of computer vision, calibration, inverse kinematics, and robotic control demonstrated reliable operation throughout experimental trials (Hastie et. al).

Performance Metric	Result
Object Detection Speed	Real-time (~25–30 fps)
Detection Accuracy (mAP)	~92% at IoU 0.5
Coordinate Error	< ±0.5 cm
Pick Success Rate	90% under stable lighting
Place Accuracy	Within ±1 cm of drop zone centre

Full Cycle Time	Approximately 10–15 seconds per object
Serial Comm Reliability	No data loss observed during testing

Table 10.2: System Performance Summary

VII. Advantages and Limitations

Advantages

- Autonomous object detection and localization
- Low-cost implementation using open-source tools
- Modular hardware and software architecture
- Real-time operation
- Accurate coordinate transformation
- Easy scalability for additional object classes

VIII. Applications

The developed system has potential applications in:

- Industrial component sorting
- Warehouse automation
- Agricultural harvesting assistance
- Pharmaceutical packaging
- Educational robotics laboratories
- E-waste segregation
- Food quality inspection
- Assistive robotic systems

IX. Conclusion

This study presented the design and development of a vision-based robotic arm integrating computer vision, deep learning, coordinate mapping, inverse kinematics, and robotic control (LeCun et. al). The system successfully demonstrated autonomous object detection and pick-and-place operations using YOLOv8 and Arduino-based actuation.

Experimental results showed a detection accuracy of approximately 92%, coordinate mapping accuracy within ±0.5 cm, and reliable robotic manipulation under controlled conditions (J. Shi et. al). The total development cost remained below ₹6,000, making the platform highly suitable for educational, research, and prototype automation applications.

The work demonstrates the feasibility of implementing intelligent robotic systems using affordable hardware and open-source software technologies (Schwab et. al).

X. Future Scope

Future developments may include:

- RGB-D camera integration for 3D localization
- Full 6-DOF inverse kinematics implementation
- Adaptive force-feedback grippers
- Dynamic lighting compensation
- Conveyor-based industrial automation
- AI-driven path planning
- Food quality inspection
- Multi-robot coordination systems

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.

[2] Z. Zhang, “A Flexible New Technique for Camera Calibration,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[3] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA, USA: O’Reilly Media, 2008.

[4] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005.

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. London, U.K.: Springer, 2010.

[6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. Hoboken, NJ, USA: John Wiley & Sons, 2006.

[7] P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, 2nd ed. Cham, Switzerland: Springer, 2017.

[8] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[9] OpenCV Foundation, “OpenCV Documentation,” 2024. [Online]. Available: https://docs.opencv.org

[10] Tzutalin, “LabelImg: Graphical Image Annotation Tool,” GitHub Repository, 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[11] Adafruit Industries, “PCA9685 16-Channel PWM/Servo Driver Guide,” 2023. [Online]. Available: https://learn.adafruit.com/16-channel-pwm-servo-driver

[12] C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[13] Autodesk Inc., “Fusion 360 for Education,” 2024. [Online]. Available: https://www.autodesk.com/education

[14] R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Cham, Switzerland: Springer, 2022.

[15] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Cambridge, MA, USA: MIT Press, 2010.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE CVPR, Las Vegas, NV, USA, 2016, pp. 770–778.

[18] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, “Visual Simultaneous Localization and Mapping: A Survey,” Artificial Intelligence Review, vol. 43, no. 1, pp. 55–81, 2015.

[19] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. New York, NY, USA: Pearson, 2018.

[20] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ, USA: Pearson, 2021
[21] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2009.
[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
[23] J. Shi, J. Wan, H. Yan, and H. Suo, "A Survey of Cyber-Physical Systems," in Proc. Int. Conf. Wireless Communications and Signal Processing (WCSP), Nanjing, China, 2011, pp. 1–6.
[24] K. Schwab, The Fourth Industrial Revolution. Geneva, Switzerland: World Economic Forum, 2016.