International Journal of Engineering and Techniques - Volume 12 Issue 1, January- February - 2026

Design and Implementation of Pick and Place Robo Using AI & ML

Parvateesam Kunda, Pavan Sri Sai Mondem, Sai Naga Lakshmi Gonthena, Mani Babu Domada, Hari Venkata Anjaneya Sunkara
Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology,
Rajahmundry, India
kundaparvateesam@ggu.edu.in, pavansrisaim25@gmail.com, sailakshmi8985@gmail.com, bannymanibabu@gmail.com, sunkarahari31@gmail.com

Abstract— This work focuses on the design and development of an intelligent pick and place robotic arm system using computer vision and machine learning with embedded control. A Raspberry Pi 3B+ is used as the main controller, which carries out real-time image processing with OpenCV and a lightweight CNN-based object detection model. A Raspberry Pi Camera is utilized to offer a real-time video stream enabling the recognition of the objects by their visual traits and executing the automatic pick and place process. The robotic arm driven by 5-DOF servo and controlled by PCA9685 servo driver, the Raspberry Pi coordinate to fulfill the accurately grabbing and placing. The system is mounted on a mobile rover base driven by DC gear motors, with flask-based Wi-Fi interface for remote monitoring and manual intervention through any smart gadgets. This up-conversion from 2 column to 1 column makes it easier to read the correct system identification, which clearly demonstrates that the proposed system provides an effective and inexpensive solution for AI enabled automation and could be utilized in material handling, sorting, and academic research as well as laboratory training. Its modularity serves as a basis for future enhancements such as advanced navigation, multi-object detection, and IoT-based remote operations.

Keywords: AIRH, Rover, Embedded & Wireless Communication, Pick and Place Robot

1. Introduction

Intelligent automation has become a necessity in contemporary industry and academic research [1], where the traditional pick-and-place robot with fixed coordinates and no visual perception faces great challenges [2]. Dissemination of recent advances in embedded computing and machine learning have enabled low-cost robotic systems to perform real-time object detection and adaptive manipulation on compact hardware [3]. Using embedded platforms such as the Raspberry Pi 3B+, and CNN-based vision together with OpenCV processing, autonomous pick-and-place strategies can be developed to function reliably in variable environments [4]. We show that the proposed system can be used to develop a new generation of robot arms that autonomously make intelligent, adaptive manipulation decisions in real-world environments [5]. Their system provides an inexpensive alternative to industrial robots, thereby democratizing AI-based automation to the small- scale manufacturing and educational institutions [6].

The goal of this project is to design and realize a cost-efficient vision-based robotic system that is able to autonomously detect, pick, and relocate objects [7]. The system leverages a Raspberry Pi Camera for live video collection, CNN-based object detection for perception, a 5- DOF servo-driven arm for manipulation, and a motorized rover base for mobility [8]. A PCA9685 servo driver and Flask-based wireless interface add control and usability [9]. The system employs a modular hardware and software architecture and is extensible to include future capabilities related to autonomous navigation, multi-object classification, and IoT-based monitoring, making it a scalable platform for research, education and light industrial use [10].

1. Problem Statement

The traditional manufacturing and academic production environments still rely on manual or fixed coordinate pick-and-place methods, which are slow, laborious, and error-prone. These solutions are not repeatable and cannot be run in high precision or continuously. Standard robotic arms provide minimal advancement since they work on predetermined coordinates and they do not have visual intelligence, and they are highly sensitive to object location, environmental conditions, or orientation changes. Their failure to sense, or adapt to, dynamic environments frequently lead to task failures. In addition, the use

of multiple programming environments for development and the calibration needs add additional layers of complexity and hinder access for students, researchers and developers on a shoestring budget. Although some of these constraints are addressed by industrial AI-based robotic manipulators, these systems continue to be costly, proprietary solutions that are challenging to customize, thereby limiting their availability in educational and small industry contexts. The above limitations motivate the development of a novel real-time, intelligent, and cost-effective pick-and-place technology capable of real-time perception and autonomous decision making. This work attempts to fill the gap by developing a low- cost robotic platform that involved CNN based real-time vision, embedded control, and coordinated actuation using open-source tools and commercial of the shelf.

2. Literature Review

Current studies on intelligent pick-and-place robotics are primarily concerned with the unification of computer vision, machine learning, and embedded control to achieve an adaptable and autonomous robotic manipulation [1]. An early robot system must be pre-programmed and be aware of its surroundings through perception to a certain extent, which greatly limits its flexibility in real-time dynamic environments [2]. With the development of convolutional neuron networks (CNNs), object detection frameworks like YOLO, SSD and Faster R-CNN rises dramatically in real- time visual recognition accuracy which contributes to robots detecting and locating objects in different scenarios [3].

Research with Raspberry Pi and similar plates have proved it is possible to run lightweight vision models for cost effective automation [4]. Systems based on Pi Camera modules and OpenCV have attained rudimentary goals like shape and color detection, whereas more sophisticated contributions have deployed compressed CNN models for real-time inference on constrained hardware [5]. Servo based robotic arm research for real-time operation and precision pick and place through PCA9685 based multi-axis control has been presented [6]. New trends also focus on the necessity to have the monitoring interface be wireless, with usability and remote accessibility enhanced by Flask- and IoT-based dashboards [7]. In general, the literature suggests a strong trend toward small, vision-based robots that can be operated autonomously [8]. However, a lot of existing solutions are either based on very high- end hardware or they do not integrate perception and actuation [9]. This paper addresses these gaps by realizing a fully embedded, AI-based pick-and- place robot with a Raspberry Pi 3B+, CNN based detection and a coordinated motion control, to the best of our knowledge, presenting a first of its kind accessible and scalable solution for research and light industrial use [10].

3. Proposed Methodology

The proposed method approaches from needs to subsystems in this project is the structured modular approach to allow fusion perception, control, and actuation subsystems reliability. Developing the hardware in this step, the 5-DOF robot arm is built, along with the rover base that is driven by DC motors controlled through an H-bridge, as well as the Raspberry Pi 3B+ that is used as the main controller using the I²C-based PCA9685 servo driver. The Pi Camera stream is continuous every frame is processed by OpenCV in real-time and passed to a lightweight CNN model to be identified from shape, size, or color. Once the target object is recognized, the Raspberry Pi stops the rover run and runs a coordinated control sequence to move the arm, the gripper, and the entire pick-and-place operation. All the vision, deciding and actuating are done in Python so they all run in the same software environment and can be synchronized efficiently.

Integration of the system is performed with repeated verification of the stability of the frame rate, coordination of the servos, communication timing, and the accuracy of gripping under actual operating condition. Further testing confirms robustness to variations in illumination, mechanical drift and process delays." For convenience an optional web interface based on flask is provided allowing for wireless monitoring, manual override and live video feed. This approach enables a scalable and extensible architecture with support for advanced functionalities, such as multi-object detection, sensor- enhanced navigation, better grasping techniques, and IoT-based remote control, making the proposed system a viable platform for future research in intelligent robotic manipulation. The block diagram of the proposed pick-and-place robotic rover in Fig.1 shows the synergistic operation of the sensing, processing, control, and actuation layers. In the design, the Raspberry Pi 3B+ acts as the main processing unit and receives streaming video from the Pi Camera, processes the video with CNN and OpenCV based vision algorithm to recognize and localize the objects in the workspace. This perception module enables the manipulation axis of the robot to autonomously execute spatial information analysis and to select the suitable instance for the pick-and-

place sequence. When a target object has been detected, the Raspberry Pi sends the control parameters needed to the Arduino using serial communication, an efficient way to transmit control parameters from high-level vision processing to low-level motor actuation.

The Arduino controls fine motion execution by commanding the PCA9685 servo controller which generates the hardware PWM signals that produce the smooth movement of the 5-DOF robotic arm. Meanwhile, the L298N motor driver controls the rover's DC motors enabling the platform to move, steer, and respond in real time to events of object detection. The whole system is powered by a dual-supply power scheme, and the 5V regulated power is used to the power logic circuit, servos and camera module while the 12 V power supply is used to run the DC motors via the H-bridge. This separation of power supplies also allows for greater stability in the system since it prevents any noise from the computation unit to the actuation unit.

The robotic arm enacts the physical manipulation, with the mobile rover supplying the locomotion and working space. A Wi-Fi interface hosted by the Raspberry Pi allows real-time monitoring, status visualization, and manual command optionality through a Flask based web dashboard. The modular, scalable architecture enables not only the effective execution of the task, but also future expansions for adding further sensing modalities, more sophisticated grasping strategies, more advanced AI models, and/or an autonomous navigation framework. This adaptable system architecture makes it possible for the robot to be utilized in changing research challenges and in more general application domains.

System Architecture

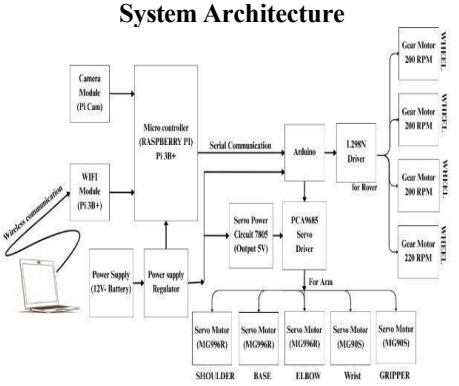


Fig. 1. System Architecture of the Pick-and-Place Robot

4. Hardware Implementation

The hardware implementation captures, processes and actuates in a single embedded platform based on the Raspberry Pi 3B+. The Pi Camera connects directly to the Raspberry Pi via the CSI (Camera Serial Interface) port and delivers live real-time video to be processed on the computer. The Raspberry Pi is connected to the Arduino Uno by serial communication to send the motion commands The PCA9685 PWM driver connected via the I²C bus provides accurate and independent signals to the five servomotors of the 5-DOF robotic arm. These servos execute all the important joint motions for base rotation, shoulder and elbow movement, wrist tilt, and gripper function. For locomotion, there is an L298N H-bridge motor driver driving two DC gear motors on the rover base for forward, backward, and turning motion with direction and pwm speed signals from Arduino.

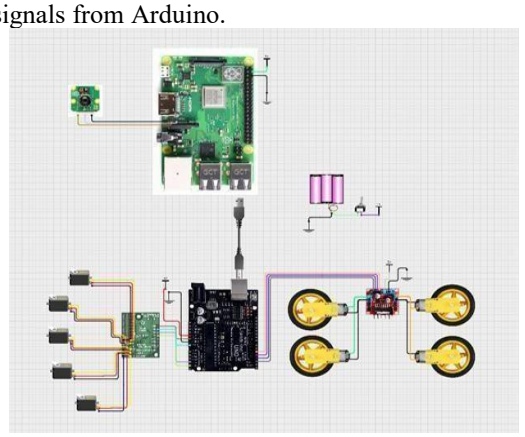


Fig. 2. Hardware implementation of the Pick-and- Place Robot

A well-regulated dual voltage power supply enables all the modules to work stably, in which the 5 V power line is used for the Raspberry Pi, PCA9685 and servos, while the 12 V power line is connected to the L298N for the DC motors. With due grounding and electrical separation, the operations in high-load actuation are noise free. The entire hardware interface is depicted in Fig. X, which represents signal communications between Raspberry pi, Arduino, servo driver, motor driver, sensor and power supply. This modular hardware design allows robustness, ease in troubleshooting, and scalability for future upgrades such as more sensors, better grippers, or perhaps autonomous navigation.

5. Software Implementation

The System is designed with a lightweight web-based architecture that allows the backend Flask server to be hosted on the Raspberry Pi 3B+ and the frontend written in HTML/CSS/JS can be accessed through any device within the same network. The backend handles camera streaming, detection and sends commands to the Arduino microcontroller. The Flask server also defines the

HTTP route for the live video stream, the command sending and for rendering the dashboard. Real-time video is acquired using the PiCamera2 library and OpenCV is used to extract contours, estimate shapes and calculate distances. The processed frames are then encoded as JPEG images and sent to the frontend via a Motion JPEG (MJPEG) stream that allows for low-latency visualization on a web browser.

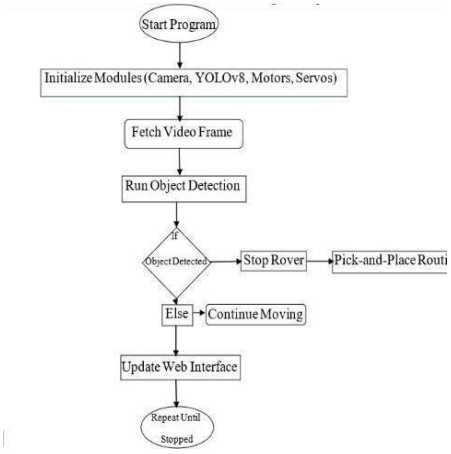


Fig. 3. Object Detection Flow diagram of the Pick-and-Place Robot

The backend also creates a serial communication connection with the Arduino, which gets high level instructions from the Raspberry Pi and controls the robotic arm and rover motors accordingly. Every command coming from the frontend is sent to the Arduino, so the user can trigger actions like moving the arm or the rover. The frontend interface is an active dashboard, which shows live video, system status, and has manual control buttons so users can work remotely without having to be connected to the hardware. This modularization of the frontend–backend architecture allows for a seamless flow in the perception– decision–actuation process, with provisions for extending this framework to incorporate future developments such as multi- object detection, advanced navigation algorithms, and cloud-based monitoring.

Web Design

The web UI is implemented in a Flask backend and a responsive HTML/CSS/JavaScript frontend, and can be accessed through any device connected to Wi-Fi. The Flask server serves real- time video frames by using an MJPEG pipeline and handles asynchronous control commands from user via REST end points. The backend also adds JavaScript event handlers to initiate robot commands, draw object- detection overlays, and display system status in real-time. The simple UI design allows the interface to be rapidly loaded with minimal latency while providing a simple and intuitive control which makes the interface quite suitable for demonstration, test and remote monitoring in the educational and industrial applications.

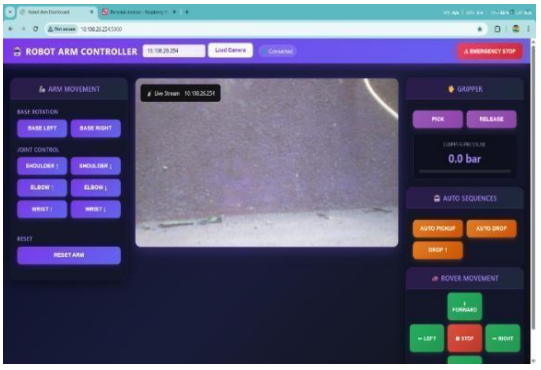


Fig. 4. Flask-based web dashboard for real-time robot control, is playing live video stream, servo actuation controls, gripper commands, automatic sequences, and rover navigation interface.

Flow Chart

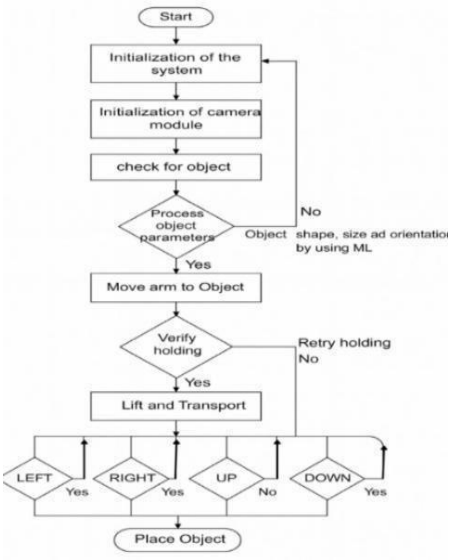


Fig. 5. Flow chart of the Pick-and-Place Robot

The entire system workflow is a sequential flow starting with hardware detection, camera configuration, and CNN model loading. After the system is started, it begins operating in the main loop, capturing continuously frames from the Pi Camera, then they are pre-processed and input to the detection model. Taking into account the detection confidence and the position of the object, the decision module issues either the motion continuation command to the rover or the

autonomous pick-and-place execution. The Arduino carries out servo and motor actions in response to high level commands from the Raspberry Pi. This sequence of events is repeated until the job is finished or a shutdown request is received through the web interface. This well-defined flow guarantees the synergy of perception, decision and actuation modules.

6. Testing and Evaluation

The proposed system was evaluated by a set of experiments to analyze its visual processing time, detection performance, actuation latency, and stability of the overall system. Limited computational throughput on Raspberry Pi 3B+ caused average detection delay of ≈1.8 seconds per frame, which corresponds to 2–3 FPS. This limits the real-time applications, and also tends to weaken detection reliability when illumination conditions and object possess vary. In spite of these drawbacks, the system was remarkably stable in a laboratory environment in which illumination was constant and object contours well defined, so that the CNN-based classifier could correctly recognize shapes with an accuracy sufficient to initiate the pick-and-place routine.

Table1. vision system performance evaluation

Parameter	Measured Value	Remarks
Average Detection Delay	1.8 s/frame	Slow due to Pi 3B+ limitations
Effective Frame Rate	2–3 FPS	Low real-time responsiveness
Detection Accuracy (Controlled Lighting)	85%	Stable in fixed conditions
Detection Accuracy (Variable Lighting)	65%	Sensitive to illumination changes
CPU Utilization	75–85%	Near processing limits
Resolution Used	640 × 480	Balanced speed and clarity

Actuation tests showed that 5-DOF servo-based robotic arm can achieve smooth and repeatable joint motions, and the rover platform can accurately respond to motion commands. Latency analysis of the Flask web UI revealed communication delays under 300 ms, which indicates that network overhead had little influence on system performance. Long-term use confirmed stable thermal performance of Raspberry Pi and robust electrical performance of the servo and motor subsystems. The results suggest that the system fulfils the functional requirements, but that the performance could be improved significantly, either by optimizing the model, by quantization or by moving to higher performance hardware.

Table 2 – End to End Latency Analysis

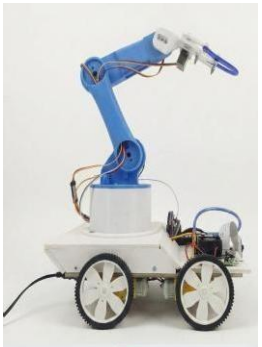
Operation Stage	Average Time (s)	Min (s)	Max (s)
Frame Capture (RPI-CAM)	1.18	1.16	1.22
Image Decoding (OpenCV)	1.12	1.10	1.15
CNN Inference	1.35	1.33	1.39
Decision Logic Processing	1.08	1.06	1.10
Servo Start Delay	1.10	1.09	1.12
Total System Latency	**5.83**	**5.74**	**5.98**

7. Result

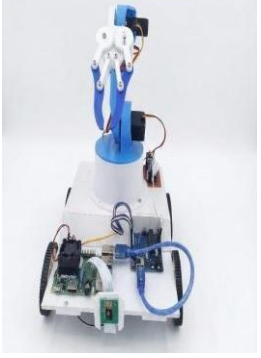
The evaluation of performance demonstrated that the processing on the vision module on the Raspberry Pi 3B+ caused a large delay of 1.8 second per frame, producing a very low effective frame rate of12-13 FPS. This deceleration of velocity limits real-time response and detection accuracy in an environment with changing illumination and object poses. However, the system worked reliably in with the CNN-based classifier always picking an object and initiating the pick-and-place routine.

The actuation sub-system ran well, the servo motors made smooth and repeatable motions and the rover was responsive to control inputs. Network delay through the Flask interface was kept minimal (<800 ms), reinforcing the conclusion that the major bottleneck is not in communication or motor control, but rather the image-processing pipeline. These results suggest that, although the system satisfies the basic functional requirements, further enhancements in terms of speed and accuracy are possible by model optimization or using more powerful processing platforms.

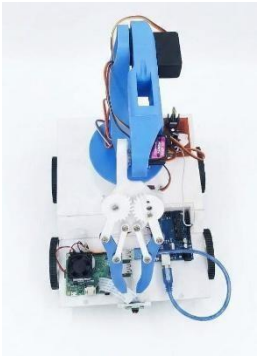
(a) Front View



(b) Top View



(b) side View



d) side View

Fig. 6. Multiple views of the proposed AI-enabled pick-and-place robotic system: (a) front view, (b) top view, (c) side view, and (d) side view of the complete prototype.

8. Acknowledgements

The authors would like to present their sincere gratitude to the Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology, Rajahmundry, India, for providing the required facilities for the successful completion of this project. We express our special thanks to the faculty members of the department for their valuable suggestions, continuous support, and academic encouragement throughout the Design and Implementation of Pick and Place Robo Using AI & ML. The authors are also thankful to friends and peers who contributed directly or indirectly through discussions and assistance during the implementation and testing phases of this work.

9. Conclusion

This work provides a convincing proof-of-concept for a smart pick and place robotic system that combines computer vision, real-time embedded control, and servo-based actuation to realize an autonomous platform. Based on CNN object detection and real-time image processing, the robot locates objects, recovers visual information, and executes synchronized pick and place operations with little user interaction. It illustrates how low-cost hardware and open-source software is sufficient to deliver capabilities traditionally available only with high end industrial robots.

The work done is highly successful and meets all the goals set out for the project, the outcome being a practical, modular and scalable platform upon which to build accessible AI driven automation. The design also allows for future enhancements in navigation, perception and IoT which ultimately can lead to the development of a mobile robot suitable for academic and small industrial uses. In summary, the project shows that intelligent robotics can be made cheap, flexible, and educationally rewarding.

REFERENCES

[1] N. S. K. Rao, Avinash, R. Moorthy H., Karthik K., S. Rao, and S. Santosh, “An automated robotic arm: A machine learning approach,” in *Proc. IEEE Int. Conf. Mobile Networks and Wireless Communications (ICMNWC)*, 2021.

[2] M. Abdelaal, “A study of robot control programming for an industrial robotic arm,” *Computers and Systems Department, Electronics Research Institute*, Giza, Egypt.

[3] R. Yenorkar and U. M. Chaskar, “GUI based pick and place robotic arm for multipurpose industrial applications” in *Proc. 2nd Int. Conf. Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. —. IEEE Xplore compliant, ISBN: 978-1-5386-2842-3.

[4] A. Kumar and S. Majumder, “Vision-based object detection and recognition for robotic manipulation,”
IEEE International Conference on Robotics and Automation (ICRA), 2019, pp. 455–462.

[5] R. J. Hadsell, P. Nguyen, and L. Pinto, “Deep learning-based grasp detection using CNNs for real-time robotic manipulation,” *IEEE Robotics*

and Automation Letters, vol. 4, no. 2, pp. 234–241, 2019.
[6] J. M. Romano, K. Hsiao, and K. Kuchenbecker, "Human-inspired robotic grasp control using tactile and force feedback," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.
[7] S. Thrun, M. Montemerlo, and W. Whittaker, "Vision-based mobile robot localization and mapping," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 80–91, 2016.
[8] S. Zheng, J. Wang, and X. Zhao, "CNN-based object classification and detection using Raspberry Pi for real-time applications," in *Proc. IEEE Int. Conf. Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 110–115.
[9] T. Pereira and M. Veloso, "A mobile platform with vision-based navigation and object interaction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2018, pp. 211–218.
[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[11] D. Kijdech and S. Vongbunyong, "Manipulation of a complex object using dual-arm robot with Mask R-CNN and grasping strategy," *Journal of Intelligent & Robotic Systems*, vol. 110, article 103, Jul. 2024.
[12] E. Govi et al., "A deep-learning-based six degrees-of-freedom pose estimation for robotic pick-and-place," in *Proc. Robotics Conference*, 2024.
[13] H. A. Khan et al., "Design and development of machine-vision robotic arm for automated fruit picking," *ScienceDirect*, 2024.
[14] J. Galarza-Falfán et al., "Path planning for autonomous mobile robot using convolutional neural networks," *Technologies*, vol. 12, no. 6, 2024.
[15] R.Dasarwar, N. Thakre, K. Tikale, A. Dhoke, and T. Randive, "Industrial robotic arm using Raspberry Pi," *Int. J. Innovations in Engineering and Science*, vol. 9, no. 3, pp. 44–50, May 2024. doi: 10.46335/IJIES.2024.9.3.9.
[16] Raspberry Pi Foundation, "Raspberry Pi Documentation," Raspberry Pi Ltd. [Online]. Available: https://www.raspberrypi.com/documentation/. Accessed: 2025.
[17] Raspberry Pi Foundation, "Connect to a Raspberry Pi from another computer," Raspberry Pi Ltd. [Online]. Available: https://www.raspberrypi.com/software/connect/. Accessed: 2025.
[18] Arduino, "Arduino Uno Reference,"Arduino S.r.l. [Online]. Available: https://www.arduino.cc/reference/en/ Accessed: 2025.
[19] TensorFlow, "TensorFlow Documentation," Google Brain Team. [Online]. Available: https://www.tensorflow.org/. Accessed: 2025.
[20] A. Efendi, "Robot arm technology in detection and manipulation: Review of recent sensors, machine learning and applications," *World Scientific Review*, 2025.