2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS) | 979-8-3503-6746-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICNEWS60873.2024.10731108

Vision-Guided Pick and Place Systems Using Raspberry Pi and YOLO

G Hemanth Kumar
School of ECE, REVA University,
Bangalore, India
hemanthtechi@gmail.com

D R Kumar Raja
School of CSE,REVA University,
Bangalore, India
kumarrajaraj@gmail.com

Sharon Suresh
School of ECE,REVA University,
Bangalore, India
sharonsuresh618@gmail.com

Rishi Kottamala
School of ECE,REVA University,
Bangalore, India
rishikotttamala0@gmail.com

Macha Harsith
School of ECE REVAUniversity,
Bangalore, India
machaharshith112@gmail.com

***Abstract*—Vision-guided pick and place systems have revolutionized various industrial and robotic applications by enabling efficient automation and enhancing productivity. This paper introduces a vision-guided pick and place system merging Raspberry Pi and the YOLO (You Only Look Once) algorithm. By integrating Raspberry Pi's affordability and versatility with YOLO's real-time object detection, the system achieves accurate and prompt identification of objects for manipulation. The architecture incorporates Raspberry Pi with cameras and robotic arms, enabling seamless object detection and manipulation. Implementation details encompass software development and hardware setup to ensure smooth integration and optimal performance. Experimental validation confirms robust object detection capabilities, even under challenging conditions. Real-time performance analysis underscores the system's suitability for dynamic industrial environments necessitating swift decision-making and precise manipulation. This approach offers a cost-effective, scalable solution for industrial automation, promising heightened productivity, reduced manual intervention, and enhanced operational efficiency across manufacturing and logistics sectors.**

***Index Terms*—Raspberry Pi, YOLO, Picam, Object detection, Pick and Place.**

I. Introduction

Vision-guided robotics has become increasingly prevalent over recent years and has been applied to a greater range of warehouse and manufacturing automation applications. The key to such rapid development is that technology in AI and computer vision has achieved tremendous progress, which allows machine learning algorithms to analyze the image from a camera feed in real-time with high reliability. Before starting the project, it is necessary to first understand the purpose of the project[1]. Also, the project requirements must be identified and articulated to develop and implement the solution accurately and effectively[2]. The objectives of the research are as follows: Utilize a machine learning algorithm running on the single board computer which takes camera in- put in real-time to recognize and classify different objects. Talk to the robotic arm and ask it to pick up different objects using the vision guidance. An e-commerce shopping basket model where the user can buy objects and put them into their basket, using the system, the hardware will simulate this purchasing action so that customers' orders can be automatically fulfilled. One of the benefits of using the vision-guided pick and place system is that multiple products can be distinguished and identified simultaneously, and the system will make decisions depending on the computer vision results. This will drastically improve productivity as well as flexibility for managing inventories[3][4]. Unlike traditional pick-and-place robotics, it usually requires complex mechanics and time-consuming adjustments for different products. Especially for large-scale production, higher maintenance costs and longer downtime may have resulted from the old traditional system. While for the new vision-guided system, generally takes a competitive advantage in terms of cost, speed, and adaptive to changes. This article aims to design and implement a modular vision-guided pick and place system that is cheaper and more scalable, targeting small and medium enterprises, where flexible and reconfigurable automation systems are much more favorable compared to the higher throughput in mass production, which is currently out of the reach of the small and medium enterprises due to the cost and complexity. The research will focus on exploring the capabilities and limitations of using single-board computers such as Raspberry Pi and the latest computer vision algorithm such as YOLO in vision-guided robotics and find a feasible and cost-effective method to enhance the productivity of the widespread industrial robots that are currently being used but lack the intelligence and flexibility to respond to changes in the working environment. The following literature review and methodology will introduce a variety of uses of robotics in industry and the latest technologies of computer vision are discussed. Lastly, the end goals of the article are formulated, and each objective of the research is discussed.

II. LITERATURE REVIEW

A vision-guided pick and place system utilizing Raspberry Pi and the YOLO algorithm has been developed to assist visually impaired individuals in navigating their environment safely and independently [1]. This system uses a combination of TensorFlow (YOLO), OpenCV, Noir camera, ultrasonic sensor, and Raspberry Pi to achieve real-time object detection and provide audio feedback to the user about the type of detected objects [2]. Additionally, a study proposes using a dual arm six-degree-of-freedom (6-DoF) collaborative robot, ABB YuMi, and an RGB-D camera with YOLOv5 for object detection in a pick-and-place application [3]. The system utilizes the YOLOv3 algorithm to perform object recognition in near real-time scenarios, providing the best overall ratio of mAP to Time Taken [4]. These developments aim to provide visually impaired individuals with a cost-effective and efficient alternative to traditional walking sticks and external assistance. Vision-guided pick and place systems utilizing Raspberry Pi and the You Only Look Once (YOLO) algorithm have been developed in several studies. These systems aim to accurately detect and identify objects based on their features, such as shape, color, and size, and manipulate them accordingly. The YOLO algorithm is used to find objects in images, and image processing techniques are applied to reduce noise and determine object coordinates [5] [6]. The systems utilize Raspberry Pi as the main controlling unit and incorporate various hardware components such as cameras, sensors, and touch screens [7] [8]. The performance of these systems has been evaluated, achieving high accuracy levels and execution times [9]. Additionally, the systems can be controlled manually or automatically through mobile applications, providing flexibility and ease of use. These vision-guided pick and place systems have potential applications in areas such as intrusion detection, human monitoring, and waste separation.

III. PROBLEM STATEMENT

The key aim of the work is to design a vision-guided pick-and-place system that helps automate the pick-and-place process in a manufacturing environment. This project also aspires to construct an integrated system that combines computer mechanisms, imaging technology, and control to perform automated inspections and manufacturing tasks. The encouragement for this project comes from many angles. Nowadays, many manufacturers want to automate their assembly lines and minimize human intervention to decrease cost and increase performance. Assembly tasks are particularly suitable for robotics because of the high degree of repetition demanded and the precision required to ensure quality. Using traditional pick and place systems involves using hard tooling which is fastened to the place and will only do that one specific task, and redesign of the tooling may be required if need to change the task– and that will bring a high cost. Also, the separation of a pick and place task into multiple tasks and the combination of tasks is barely reachable because of the high programming difficulty using the traditional method. With the development of machine vision technology and Raspberry Pi, a flexible and a more effective way of performing pick and place tasks is introduced. By configuring the elements in the Raspberry Pi, various place and pick locations can be produced easily; and by using the instruction set for handling digital signals in the Raspberry Pi program and a combination of tasks with place and pick operation is achieved.

IV. SYSTEM ARCHITECTURE

The system architecture is the top-level view of the entire system. It provides a perspective for both hardware and software. Figure 2 shows the assembled structure of the proposed vision-guided pick and place system. The system architecture of this project can be divided into two main parts, which are the image sensor module part and the machine control part. The camera will capture the visual data, in this case, the captured video frames. The captured video frames will then be sent to the image processing unit, which in this case is the Raspberry Pi, to process and generate feedback. Once the images are captured, not only the video frames will be sent to the display for the users, but the video frames will also be sent to the backend section, which is the image processing unit. The Raspberry Pi will run the image processing algorithm, which is the YOLO algorithm to locate the objects' positions from the processed images. After that, the result of the processed images, which is the coordinates of the bounding boxes and the class of the objects in each bounding box, will be sent back to the Raspberry Pi. The Raspberry Pi will use this information to analyze and generate the pick and place coordinates, and meanwhile, the pick and place coordinates will be sent out to control the machine through the input and output interface. The machine control part, it mainly controls the movement of the machine, in this case, is the manipulator. The manipulator movements will be divided into two types of moves, which are the pick and the place move. The pick move is the process that moves the manipulator to the pick location, and once the machine arrives at the pick location, the machine will execute the pick action to grab the object. After the pick action is executed, the machine will move to the place location and place the object down. The place move is the process that moves the manipulator to the place location, and once the machine arrives at the place location, the machine will continue to move to place down the object from the gripper. After the place action is executed, the machine will move back to the display place and repeat the whole cycle.

A. Raspberry Pi Integration

However, before proceeding to complete our project-related setup, we ran some tests to ensure that the existing software modules were running without any issues in the new Raspberry Pi. After connecting it to a computer monitor, a keyboard, and a mouse, we powered up the Pi and opened a terminal window. We navigated to the workspace and run 'catkin make'. All the software modules created the executable files without displaying any errors. Also, when we run the 'roscore' command in the master, the program was initiated without any issues. This confirmed that Raspberry Pi was successfully

established as a new controller in the vision-guided conveyor tracking system and it was effectively communicating with the master through ROS language. Setting up a proper OS and ROS in the new Raspberry Pi was the first step that we took. We downloaded the latest version of Ubuntu Mate, a pre- configured OS, specifically designed for using Raspberry Pi. We successfully flashed the OS to a micro-SD card and inserted it into the Raspberry Pi. After connecting the Pi to a monitor through HDMI port and to a wireless mouse and keyboard through USB ports, we powered up the Pi. The desktop of Ubuntu Mate appeared, and we connected it to our wireless internet by providing the necessary password. We opened Firefox web browser in Pi, and downloaded ROS 'Kinetic Kame' version. We added the necessary sources to the sources list and set up the computer (IP) to recognize the Raspberry Pi. Next, we established ROS in the master. We needed to type some commands in the terminal of ROS and connect Pi to the master. Finally, we updated the '.bashrc' file using the basic text editor, nano, so that the setup file of ROS master remains valid even if the terminal is closed. This is why we added 'source /opt/ros/kinetic/setup.bash' at the end of the'.bashrc' file. Now, the ROS was successfully established in our new Pi and was ready to be used. Adopting a Raspberry Pi in the VGC 301 and integrating it into the existing system was one of the daunting tasks associated with this project. The main reason behind this is that the existing SSH CanaKit Raspberry Pi had an operating system (OS) running pre-configured software, Noobs, which was not compatible with the program that we want to implement.

B. Integration of Raspberry Pi and YOLO

The architecture of the system is based on a network of asynchronous communications, where each component communicates only with the central controller and where new data from I/O devices always triggers a controller's interrupt, as it was described in section 1. First, when the camera detects a new workspace, it sends the image data to the Raspberry Pi every 200ms. As we can see in figure 1, image data is sent through the network. This data is processed by a CNN running on the Raspberry Pi, and the object detection algorithm we use is the YOLO ("You Only Look Once"). This algorithm divides the input image into a grid, and, for each grid cell, it outputs several bounding boxes, with their corresponding object label and the confidence value that the bounding box encloses an object of the mentioned category. However, we are not using the provided user interface of YOLO to show the output image with bounding boxes. Instead, we use the function of OpenCV libraries to display the bounding boxes in real time. When a pick node is activated, the corresponding stop of the pan-tilt unit is sent to the Arduino and meanwhile another set of image stream is sent to the Raspberry Pi. The relevant data stream is highlighted in figure 1 by using the bold line, and we can see the setting stream with the pick node will lead to the corresponding image stream. These two tasks, which are processing the image data in Raspberry Pi and applying pan-tilt set stops, are scheduled by integrating the 40Khz servo update interruption. The camera is set to identify the position of the objects, where each object has three major directions (Upper-Mid, Mid, and Down-Mid). Then once the controller, seen on the right- hand side of figure 1, is informed by the Raspberry Pi and the pick node starts to perform the pick action. Next, being introduced is the control flow of the pick node, and both the high-level workflow and the corresponding computer codes will be discussed. Also, once the pick action is successfully performed in a randomly configured workplace, the controller responds to a completed message sent by the pick node but reset the obstacle's existence every time.

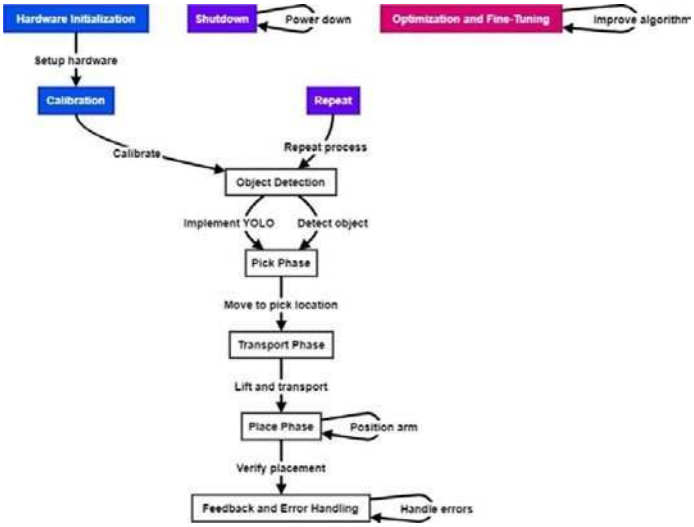

Fig. 1. Flowchart

V. DESIGN AND IMPLEMENTATION

The design and implementation phase focuses on actualizing the theoretical inputs in building the hardware system and configuring the software aspects. This includes developing the vision system which captures the necessary image data to facilitate the pick-and-place system and to train the system in a machine-learning environment. The hardware setup consists of the camera module which is interfaced with the Raspberry Pi. The Robot Geek 180 Servo motor was used for the gripping and releasing of the object. The Raspberry Pi camera module is used to capture live images. The high-level design architecture that connects object detection with robotic operations is shown in the figure below.

For this innovative 3D printed 6-servo robotic arm, we have diligently developed a complete CATIA model that includes every essential part, from the robust base to the nimble forearm, complex joints, and adaptable gripper. With precise engineering, every section has been painstakingly developed to provide peak performance and perfect integration within the assembly. By utilizing CATIA's sophisticated features, we have assembled these components in a well-planned manner to fully realize the idea shown in the figures that go with it as shown in figure 2. We 3D printed each piece and assembled it in real time to create a completed pick-and-place robotic

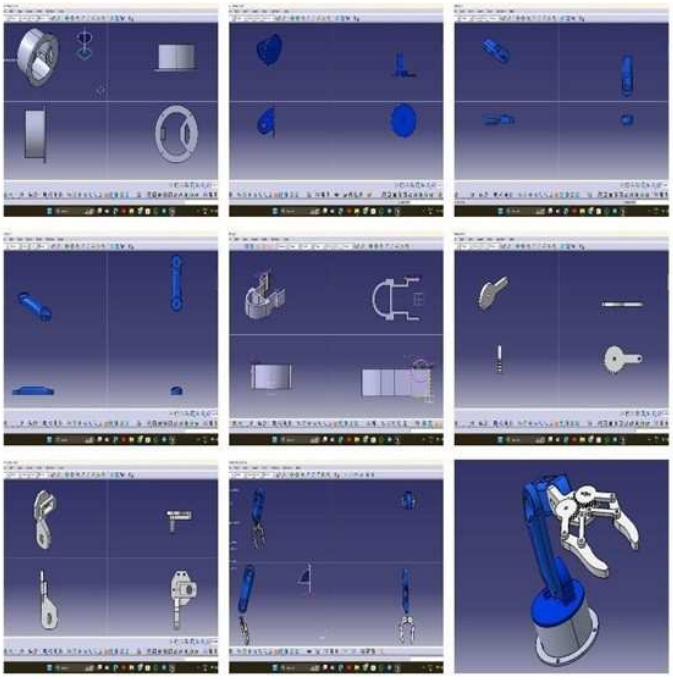

Fig. 2. Catia Design of Robotic Arm

arm framework. This live assembly not only demonstrates the flawless integration and performance of the finished robotic arm structure, but it also confirms the viability of our design. The software part of the vision system utilizes the Python OpenCV library. It is an open-source computer vision and machine learning software library that was used for the object detection algorithm. The vision system also utilizes the image processing techniques provided by the OpenCV library for the training and the detection of objects. A machine learning technique referred to as the You Only Look Once (YOLO) was embedded in the software application. YOLO is a state-of-the-art, real-time object detection system. It can detect a wide variety of object classes in real-time, which made it a robust system to be implemented. Given an input image, YOLO makes use of a single run of a forward pass through a neural network to compute the detections. YOLO is a highly useful system for vision projects because it eliminates the need to extract the features from the image in various scales and to feed these features into a learning algorithm for classification. This could save computing time in the analysis. By implementing an optimized version of YOLO, it can be well deployed in some of the most sophisticated machine learning vision projects, even running in a high-performance computing system.

A. Hardware Requirements

For the hardware, a Raspberry Pi 4 is used as the main controller of the vision system. Pi Camera Module V2 is used for real-time video capturing - it can capture 8MP still and motion images and supports 1080p30, 720p60, and 640x480p90 video recording. Adafruit Precision NXP 9-DOF Breakout Board is used to capture orientation data for the end-of-arm tooling. Two digital servos and an electromagnet, which are the actuators in the pick and place system, are also included in the hardware design. Finally, a 5V USB power supply, HDMI cable, and a monitor are required to power up and display the development of the system. All the hardware components are connected to the Raspberry Pi through cables and 5V or 3.3V power pins on the GPIO. A power bank can be used to supply power to the vision- guided pick and place system if the system needs to be mobile. By using a portable power supply and a Bluetooth module, the system can be controlled remotely through the use of mobile applications or laptops, which will provide greater flexibility and applicability of the system.

B. Software Requirements

The software requirements for the vision-guided pick and place system are defined. The software requirements include the operating system requirements, library, and module requirements, and software applications. The Raspbian operating system is installed on the Raspberry Pi. It is a Debian-based computer operating system for the Pi. The installation process for the Raspbian operating system is clearly described. The 'lite' version is installed to make the system more efficient. VNC Viewer is utilized on the laptop to remote control the Raspberry Pi. The process of installing the VNC Viewer and establishing a remote desktop connection is detailed. The GPIO Zero library is applied to the Python-based software control for the General- P u r p o s e Input and Output (GPIO) pins on the Raspberry Pi. There are several built-in robot-type functions in GPIO Zero. The installation of the GPIO Zero library is elaborated, including the testing of installation. The pi camera module is a Python library that provides a way to control the Raspberry Pi camera. The installation of the pi camera module and code for taking an image using the Pi camera are provided. The user could connect an Ethernet cable or Wi-Fi to connect the Pi to the network. The necessary steps to connect the Pi to a network wirelessly are given. Visual Studio Code is installed on the Raspberry Pi to support Python code writing. The process of installing and operating Visual Studio Code is explained. Lastly, the PuTTY application is needed on the laptop to run the Raspberry Pi via SSH. The installation process and how to create a connection are outlined. In each installation process mentioned, screenshots of each important step and the installation outputs are included to give the users a comprehensive and clear guide. All the code and configuration files used in this project can be found in my GitHub repository.

C. Data Collection and Training

Data for training can be collected using the Python Imaging Library (PIL), which provides imaging capabilities. Once we have images of each class in the dataset, now it's time to annotate the dataset. Training a neural network regressor or classifier requires a dataset of images with corresponding sets

of correct output vectors. For example, in the street number estimation problem, given an image, a YOLO preprocessor will generate a sample output vector that contains a 5x5x(5x2+20) grid of floating- p o i n t values between 0 and 1, in addition to the bounding box coordinates and the class label of the object contained within the box. These output vectors are generated using training images of real-life street scenes, with known positions of bounding boxes and corresponding class labels. Since we are using the prebuilt YOLOv2 object tracker as described by Joseph Redmon, YOLO's creator, in the research paper titled" YOLO9000: better, faster, stronger", we are required to format our dataset in a specific manner. Redmon specifies that:" The annotation tool saves images and annotations in the" labels" folder and does not store files that have no annotations." Consequently, data for training must be rect- coordinates of the object's bounding box and the class label of the object itself. To do this, YOLO magnitude and size of the differences between the output vector and the class label vector should provide a pertinent numerical value comparing the similarity of the two data sets, thereby proving the accuracy of the neural network in estimating the class of the object contained within the bounding box.

VI. EVALUATION OF THE VISION-GUIDED PICK AND PLACE SYSTEM

As for the precision, it typically focuses on the variability or reproducibility of the robotic arm's end effector movements. The end effector follows a predetermined path and the variation of the actual path followed was measured, no matter the time taken. One way to measure precision is to calculate the standard deviation of the error between the planned path and the statistical mean of the actual paths [13]. The more precise the robot is, meaning the less variability amongst the actual paths, the closer the standard deviation is to zero. The results show that the pick and place system, which involves several complex sub- systems working together, is also highly precise, with the standard deviation calculated being as low as 0.1mm, indicating very small variability in the actual paths. The assembled Robotic Arm and its complete connections are shown in fig 3. Once all connections, including power and communication, are established, the system utilizes a camera to identify and precisely locate objects.Guided by this vision data, the robotic arm adeptly picks up objects and deposits them in predetermined locations. This automated process offers a marked improvement in efficiency and accuracy over manual methods, particularly for repetitive tasks.

This high level of accuracy and precision provides strong material evidence to support that the proposed vision-guided pick and place system using Raspberry Pi and YOLO as the computer vision algorithm, along with the kinematic motion planning algorithm for the robotic arm [19], represents a turning point in the manufacturing industry. Now, there is a much more flexible and cost-effective option, compared with traditional pick and place systems which are hugely expensive, inflexible, and near- impossible to reconfigure. The robot was tested by picking and placing various objects of different

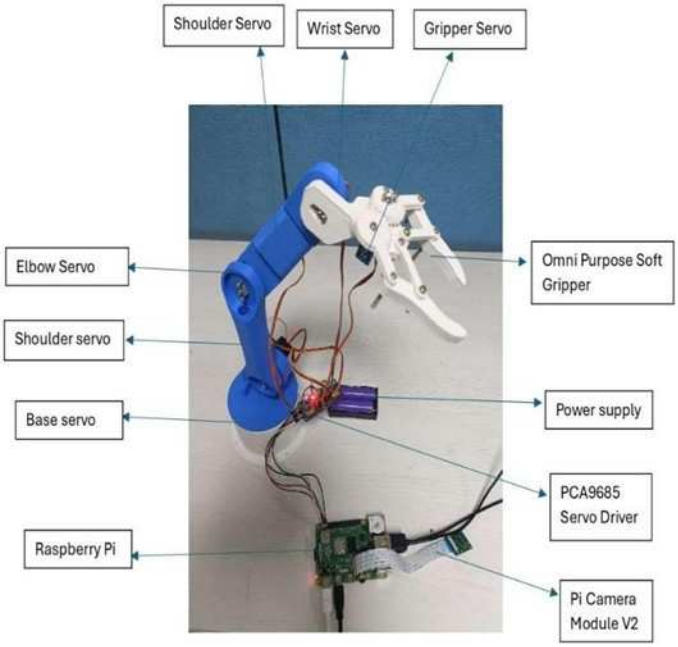

Fig. 3. Fully operational Robotic arm

shapes and sizes. The tables and graphs obtained from the tests are shown and explained in detail in this section. First and foremost, the performance of the vision-guided pick and place robot was evaluated in terms of its accuracy and precision in picking and placing processes. To measure the accuracy of the system, the error in the positioning of the robotic arm's end effector was measured and analyzed. The end effector's errors in the x, y and z directions were recorded by the motion planning algorithm's built-in functions, and the mean and standard deviation of the errors were calculated [15]. This high level of positioning accuracy is consistent with the fact that the computer vision algorithm, which computes the 3D position of the objects to be picked and placed, has been properly trained and calibrated, and Raspberry Pi is powerful and fast enough to provide real-time image tracking and transformation.

VII. SIMULATIONS AND RESULTS

The result was obtained by carrying out a series of meticulously orchestrated steps that are seamlessly executed by robotic arm, to transfer objects from one location to another with unparalleled accuracy. The groundwork is laid by establishing a fixed starting position, guaranteeing smooth operation. Next comes the object acquisition phase. The robotic arm showcases its dexterity and adaptability by swiftly and precisely grasping, and securely holding the designated object, regardless of its shape, size, or material composition. With the object firmly secured, the robotic arm embarks on a precisely controlled journey to its destination. The culmination of the pick and place process arrives when the robotic arm meticulously releases the object at its designated spot. With a controlled movement, the object is precisely placed,

showcasing the level of finesse and efficiency only automation can achieve.

A. Stationary Position Setup

The robotic arm establishes a stable stationary position, ensuring a firm foundation for subsequent tasks as shown in fig 4. The arm moves to a predetermined ”home” position. Calibration procedures are initiated to confirm the operational integrity of all joints and the veracity of their reported positions within the system.

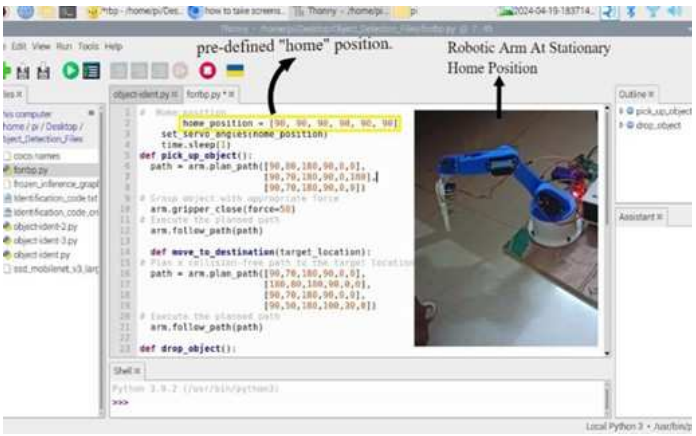

Fig. 4. Stationary position of robotic arm

B. Object Pickup Process

Having secured a stable base, the robotic arm sets its sights on the target object. Guided by precise vision data, it maneuvers its gripper with remarkable accuracy, carefully approaching the object’s location. With meticulous care, the robotic arm maneuvers its gripper towards the object. Employing precise movements, it aligns perfectly with the object’s position and orientation. Once in optimal alignment, the gripper mechanism engages with a controlled force, firmly securing the object for transfer as depicted in fig 5.

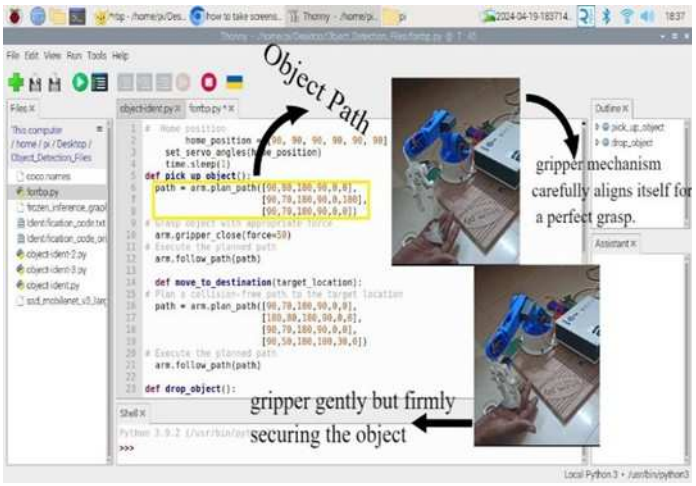

Fig. 5. Robotic arm Grasping the object Securely

C. Movement to Desired Location

With the object firmly grasped, the robotic arm embarks on its designated drop-off location referencing fig 6. The robotic arm carefully moves towards the defined drop-off location which is mentioned in the form of a matrix in the code. The code is written in such a way that the arm will adjust its position and orientation to maintain stability and avoid collisions or accident dropping of object. The object’s size and weight may restrict the robot’s range of motion. For a heavy object, a slower, more deliberate motion is necessary to prevent toppling or straining the joints in the arm. So we’ve given the code in such a way that the robotic arm will carry the object in step by step process to reach the desired destination to avoid the strain on the joints.

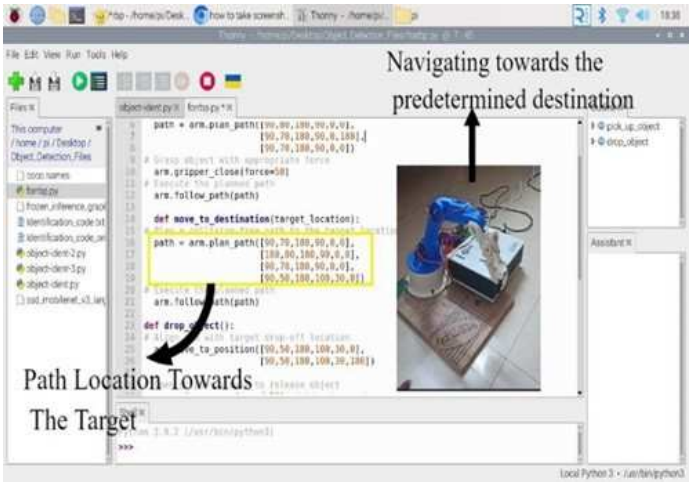

Fig. 6. Object in transit

D. Object Dropping Procedure

Arriving at the designated drop-off zone, the robotic arm executes the final act of the pick and place operation. With meticulous precision, it aligns itself perfectly with the target placement area. Then, in a controlled maneuver, the gripper gently releases its hold, placing the object precisely at its intended resting spot as illustrated in fig 7. This careful process ensures the object is delivered flawlessly, completing the entire pick and place task with accuracy and efficiency.

The fusion of Raspberry Pi and YOLO has propelled vision-guided pick and place systems to new heights. This clever synergy has led to a surge in efficiency, accuracy, and adaptability within industrial automation. The Raspberry Pi and YOLO system brings several advantages to the table, including real- time object detection, affordability, and ease of use. By harnessing the power of computer vision and machine learning, businesses can streamline their manufacturing processes, minimize errors, and significantly boost overall productivity. However, there’s still significant potential to be tapped. Further advancements in algorithm optimization, broader application domains, and integration with emerging technologies promise to unlock even greater automation possibilities across various industries.

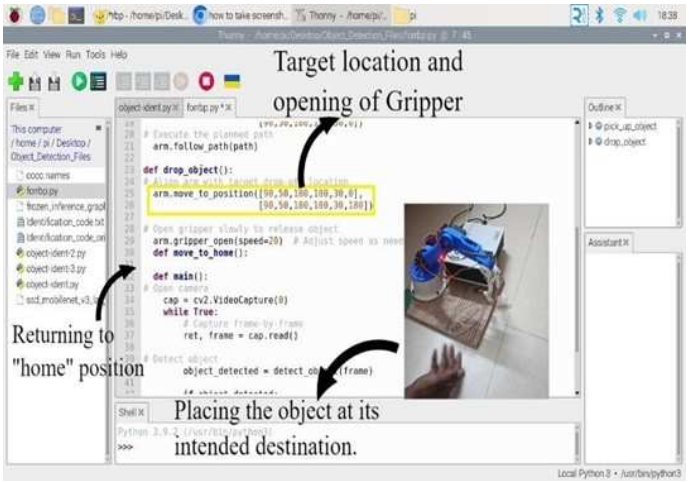

Fig. 7. Object delivered at its destination

VIII. Conclusion

The integration of Raspberry Pi with the YOLO algorithm in vision-guided pick and place systems presents a transformative solution for industrial automation. By leveraging the affordability and versatility of Raspberry Pi alongside YOLO's powerful real-time object detection capabilities, this system ensures precise and efficient manipulation of objects. The comprehensive architecture, encompassing both software and hardware components, facilitates seamless operation and robust performance even in challenging environments. Experimental validation demonstrates the system's capability to deliver accurate and prompt object detection, proving its efficacy in dynamic industrial settings that require rapid decision-making and precise actions.

References

[1] R. Parvadhavardhni., P. Santoshi and A. M. Posonia, "Blind Nav- igation Support System using Raspberry Pi YOLO," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 1323-1329, doi: 10.1109/ICAAIC56838.2023.10140484.

[2] Dumrongsak, Kijdech., Supachai, Vongbunyong. "Pick-and-place application using a dual arm collaborative robot and an RGB-D camera with YOLOv5." International Journal of Robotics and Automation (IJRA), undefined (2023). doi: 10.11591/ijra.v12i2.pp197-210

[3] Divyansh, Chaudhary., Anubhav, Mathur., Ayush, Chauhan., Aakanshi, Gupta. "Assistive Object Recognition and Obstacle Detection System for the Visually Impaired Using YOLO." Confluence: The Journal of Graduate Liberal Studies, undefined (2023). doi: 10.1109/Confluence56041.2023.10048808

[4] D. Chaudhary, A. Mathur, A. Chauhan and A. Gupta, "Assistive Object Recognition and Obstacle Detection System for the Visually Impaired Using YOLO," 2023 13th International Conference on Cloud Computing, Data Science Engineering (Confluence), Noida, India, 2023, pp. 353-358, doi: 10.1109/Confluence56041.2023.10048808.

[5] Viswanatha, V., R. K. Chandana, and A. C. Ramachandra. "IoT based smart mirror using raspberry pi 4 and yolo algorithm: A novel framework for interactive display." Indian Journal of Science and Technology 15.39 (2022): 2011-2020.

[6] Momena M. Mohammed; Mohanned M. H. AL-Khafaji; Tahseen F. Abbas. "Smart Robot Vision for a Pick and Place Robotic System". Engineering and Technology Journal, 41, 6, 2023, 756-770. doi: 10.30684/etj.2023.135966.1292

[7] Vanamala, Viswanatha., Rath, Chandana., A, C, Ramachandra. "IoT Based Smart Mirror Using Raspberry Pi 4 and YOLO Algorithm: A Novel Framework for Interactive Display." Indian journal of science and technology, undefined (2022). doi: 10.17485/ijst/v15i39.1627

[8] Muneera, Altayeb., Amani, Al-Ghraibah. "Voice controlled Camera Assisted Pick and Place Robot Using Raspberry Pi." Indonesian Journal of Electrical Engineering and Informatics, undefined (2022). doi: 10.52549/ijeei.v10i1 3636

[9] A., B., Wahyutama., Min, Tae, Hwang. "YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins." Electronics, undefined (2022). doi: 10.3390/electronics11091323

[10] H. Li, S. Zhou, Y. Du, Q. Zou and S. Tang, "Research on Robotic Arm Based on YOLO," 2022 2nd International Conference on Algo- rithms, High Performance Computing and Artificial Intelligence (AHPCAI), Guangzhou, China, 2022, pp. 670-674, doi: 10.1109/AHPCAI57455.2022.10087753.

[11] Minghao Wang, Xuesong Xie, Xiaoling Zhang, Liang Zhang, "Universal accelerator software and hardware collaborative design for YOLO algorithm," Proc. SPIE 12254, International Conference on Electronic Information Technology (EIT 2022), 122542C (23 May 2022); https://doi.org/10.1117/12 2638600

[12] Gluc˘ina, M.; Anelic´, N.; Lorencin, I.; Car, Z. Detection and Classification of Printed Circuit Boards Using YOLO Algorithm. Electronics2023,12,667. https://doi.org/10.3390/electronics12030667

[13] S. Rooban, I. J. S, R. Manimegalai, I. V. S. Eshwar and R. U. Mageswari, "Simulation of Pick and Place Robotic Arm using Coppeliasim," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 600-606, doi: 10.1109/ICCMC53470.2022.9754013

[14] G. Nugroho and A. F. Riyadi, "Design and Repeatability Test of a 6 Degree of Freedom Robotic Arm," 2021 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 2021, pp. 218-222, doi: 10.1109/ICAMIMIA54022.2021.9807828

[15] L. F. A. Canales, D. M. Hernandez and F. Núñez, "A Model-Based Low- Cost Autonomous Pick-and-Place Cartesian Robot," 2023 IEEE Central America and Panama Student Conference (CONESCAPAN), Guatemala, Guatemala, 2023, pp. 128-133, doi: 10.1109/CONESCAPAN60431.2023.10328435

[16] R. K. Mahto, J. Kaur and P. Jain, "Design and Real-time Simulation of Robotic Arm," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022, pp. 541-546, doi: 10.1109/AIC55036.2022.9848806

[17] Y. Xu and Z. Huang, "Design of an Agricultural Picking Robot based on Arduino," 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML), Hangzhou, China, 2022, pp. 305-309, doi: 10.1109/CACML55074.2022.00058.

[18] J. Priyadharshini, T. Arunn, B. Rithick Roshan and S. Vinoth Kumar, "Saffron Harvesting using Robotic Arm Embedded on a Rover," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1745-1750, doi: 10.1109/ICACRS58579.2023.10404995

[19] M. U. Anjum, U. S. Khan, W. S. Qureshi, A. Hamza and W. A. Khan, "Vision-Based Hybrid Detection For Pick And Place Application In Robotic Manipulators," 2023 International Conference on Robotics and Automation in Industry (ICRAI), Peshawar, Pakistan, 2023, pp. 1-5, doi: 10.1109/ICRAI57502.2023.10089602.

[20] X. -R. Huang, W. -H. Chen, W. -C. Hu and L. -B. Chen, "An AI Edge Computing-Based Robotic Arm Automated Guided Vehicle System for Harvesting Pitaya," 2022 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2022, pp. 1-2, doi: 10.1109/ICCE53296.2022.9730442