

Article

Robotic Harvesting of Apples Using ROS2 [†]

Connor Ruybalid, Christian Salisbury and Duke M. Bulanon *

Department of Engineering and Physics, Northwest Nazarene University, Nampa, ID 83686, USA; cruybalid@my.nnu.edu (C.R.); christiansalisbury@my.nnu.edu (C.S.)

* Correspondence: dbulanon@nnu.edu

[†] This paper is an extended version of our published paper: Ruybalid, C.; Salisbury, C.; Bulanon, D.M. Robotic Fruit Harvesting with Dual Camera System. In Proceedings of the 2025 IEEE Opportunity Research Scholars Symposium (ORSS), Atlanta, GA, USA, 22–24 April 2025; pp. 1–4.

Abstract

Rising global food demand, increasing labor costs, and farm labor shortages have created significant challenges for specialty crop production, particularly in labor-intensive tasks such as fruit harvesting. Robotic harvesting offers a promising long-term solution, yet its adoption in orchard environments remains limited due to unstructured conditions, variable lighting, and difficulties in fruit recognition and manipulation. This study presents an improved robotic fruit harvesting system, Orchard roBot (OrBot), developed by the Robotics Vision Lab at Northwest Nazarene University, with the goal of advancing autonomous apple harvesting applications. The updated OrBot platform integrates a dual-camera vision system consisting of an eye-to-hand stereo camera with a wide field of view for fruit detection and an eye-in-hand RGB-D camera for precise manipulation. The control architecture was redesigned using Robot Operating System 2 (ROS2) and Python, enabling modular subsystem development and coordination. Fruit detection was performed using a YOLOv5 deep learning model, and visual servoing was employed to guide the robotic manipulator toward the target fruit. System performance was evaluated through laboratory experiments using artificial trees and field tests conducted in a commercial apple orchard in Idaho. OrBot achieved a 100% harvesting success rate in indoor tests and a 75–80% success rate in outdoor orchard conditions. Experimental results demonstrate that the dual-camera approach significantly enhances fruit search efficiency and harvesting efficiency. Identified limitations include sensitivity to lighting conditions, end effector performance with varying fruit sizes, and depth estimation errors. Overall, the results indicate a positive potential toward effective robotic fruit harvesting and highlight key areas for future improvement in vision, manipulation, and system robustness.

Keywords: apple; harvesting; machine vision; robotics



Academic Editors: Raul D. S. G. Campilho and Marco Ceccarelli

Received: 12 February 2026

Revised: 2 April 2026

Accepted: 11 April 2026

Published: 14 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Global food systems are facing increasing pressure from population growth, rising production costs, climate variability, and persistent labor shortages. It is projected that global food production will need to double by 2050 to meet the demands of a population expected to reach nearly 10 billion [1]. At the same time, agriculture—particularly specialty crop production—remains heavily dependent on manual labor for tasks such as harvesting. In the United States, and especially in Idaho, one of the nation's top five producers of specialty crops, growers of fruits and vegetables are increasingly challenged by a shrinking

labor force and rising wages. Although the H-2A Temporary Agricultural Worker Program [2] has helped mitigate labor shortages, it represents a short-term and increasingly costly solution, highlighting the need for sustainable, long-term alternatives.

Technological advancements in precision agriculture, automation, and robotics offer promising pathways to address these challenges [3,4]. While robotics has been successfully integrated into highly structured environments such as automotive manufacturing, its adoption in agriculture—particularly in orchards—remains complex. Orchards present unstructured and dynamic environments characterized by variable lighting, occlusion from foliage, and natural variability in fruit location and tree geometry. These factors make robotic harvesting significantly more challenging than industrial automation, where workpieces are consistently positioned, and lighting conditions are controlled [5].

Despite these challenges, robotic fruit harvesting has been an active area of research for more than three decades. Early systems demonstrated feasibility using black-and-white cameras and simple robotic manipulators, and subsequent studies have explored a wide range of sensing technologies, including color, stereo, thermal, multispectral, and depth cameras [6]. Among the key technical challenges are reliable fruit recognition, precise visual servoing of robotic manipulators, and the development of end effectors capable of removing fruit without damaging either the produce or the tree [7]. Fruit recognition, in particular, has received significant research attention, as accurate detection is a prerequisite for successful harvesting [8].

The Robotics Vision Lab of Northwest Nazarene University has been developing a robotic harvesting platform, Orchard roBot (OrBot), to investigate these challenges in real-world orchard environments [9]. OrBot integrates robotic manipulation with machine vision systems to locate and harvest apples and has demonstrated promising results in commercial orchards. Although its harvesting speed remains slower than that of human labor, robotic systems offer the potential to operate continuously and complement human workers, including during nighttime hours when labor is unavailable. This study builds on previous iterations of OrBot [10] by exploring improved vision systems, control systems, and harvesting strategies, with the goal of advancing robotic fruit harvesting toward a more practical and economically viable solution to the growing labor challenges faced by specialty crop producers. The first iteration of OrBot used MATLAB (R2022b) to develop its control system, and it used only one camera for finding and locating the fruit.

The objectives of this study are:

1. To improve the control system of OrBot using Robot Operating System 2 (ROS2);
2. To improve the fruit vision system by adding another camera with a wider field of view;
3. To evaluate the performance of OrBot in fruit harvesting.

2. Materials and Methods

2.1. Fruit Harvesting Robot: Orchard Robot (OrBot)

For the study of robotic fruit harvesting, the Robotics Vision Lab of Northwest Nazarene University developed a robotic platform called Orchard Robot, OrBot. OrBot is composed of the following: the manipulator, the eye-in-hand vision system, the eye-to-hand vision system, the end effector, a control system, a power supply, and a tank-tread-based navigation platform (Figure 1).

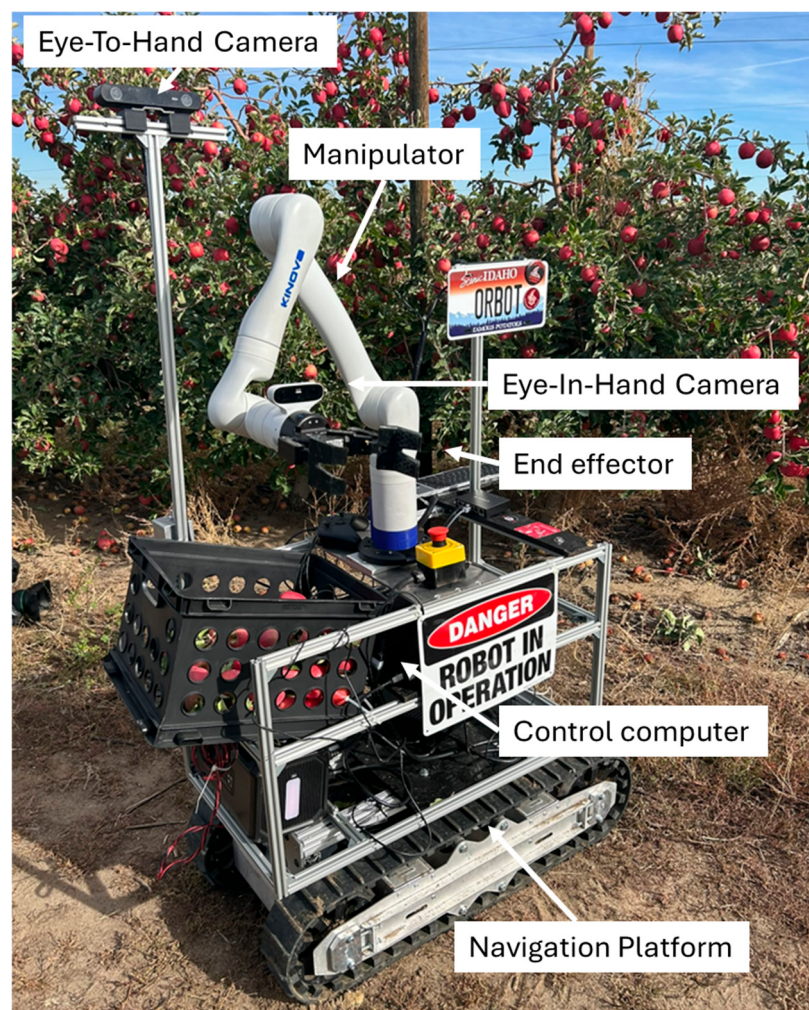


Figure 1. OrBot and its components.

2.1.1. Manipulator

The manipulator is a third-generation Kinova robotic arm (Kinova Robotics, Boisbriand, QC, Canada) [11]. This arm has six degrees of freedom and has a reach of 902 mm. The arm has a full-range continuous payload of 2 kg, which is more than the weight of an apple fruit, and it has a maximum Cartesian translation speed of 500 mm/s. The average power consumption of the arm is 36 W.

2.1.2. Eye-in-Hand Vision System

The first version of OrBot has only one vision system [10] and is referred to in this paper as the eye-in-hand vision system. This vision system is integrated with the end effector. The vision system, composed of the color sensor, has a resolution of 1920×1080 pixels with a field of view up to 65 degrees. The depth sensor, which is enabled by both stereo vision and active IR sensing, has a resolution of 480×270 with a field of view of 72 degrees. The color sensor is used to detect the fruits, and the depth sensor is used to estimate the distance of the fruit from the end effector.

2.1.3. Eye-to-Hand Vision System

In the current version of OrBot, another vision system was added [12]. This vision system is fixed and does not move with the end effector. In this paper, we refer to this vision system as the eye-to-hand vision system. The ZED 2 Stereo Camera (Stereolabs, San Francisco, CA, USA) is used as the eye-to-hand vision system [13]. Unlike the eye-in-hand

vision system, the ZED 2 camera has a wider field of view (FOV) of 110° (H) × 70° (V) and a resolution of 2× (1920 × 1080). It has a depth range of 0.3 m to 20 m.

2.1.4. End Effector

The end effector is a standard two-finger gripper. The gripper has a stroke of 85 mm and a grip force of 20 to 235 N, which could be controlled depending on the target. Customized 3D-printed fingers of the gripper were designed and developed to allow for a larger stroke for larger-sized fruit and to conform to the spherical shape of the fruit.

2.1.5. Control System

The control system used ZED Box Orin NX (Stereolabs, San Francisco, CA, USA) is powered by an NVIDIA Jetson Orin NX with an Ampere GPU, Arm Cortex CPU, and LPDDR5 memory, as well as multiple USB ports, Ethernet, HDMI, and ports for cameras [14]. The control system used UBUNTU as the operating system and Python 3.10 as the main programming language. A foldable keyboard and monitor were mounted to the chassis of the robot for easy access in the field.

2.1.6. Tank-Tread Navigation Platform

The tank-tread navigation platform is an HD2 Treaded ATR Tank Robot Platform [15]. It is driven by 2 24 VDC Gear Motors with encoders and has a payload capacity of 45 kg. The platform has been configured to be able to mount OrBot, its components, the power supply, and other materials as needed.

2.2. Object Detection Algorithm

YOLO (You Only Look Once) is a series of real-time object detection models built on PyTorch 2.0.0 and developed by Ultralytics [16]. This project used YOLO version 5 (YOLOv5). As a brief overview, the architecture of YOLOv5 contains a backbone, neck, and head. The backbone is the main body of the model, designed using the CSPDarknet53 structure, which is a convolutional neural network. The head is in charge of generating a final output. The neck connects the backbone and head.

YOLOv5 provides tools and Python scripts for training the model on a custom dataset, changing the weights to serve custom purposes. These new weights can then be loaded into the YOLOv5 architecture and used to perform inference.

When evaluating an object detection model like YOLO, it is common to use the mean average precision. The average precision refers to the average precision level of the model over every recall level. Precision measures the accuracy of the positive predictions made by the model, calculated as the ratio of true positives to the sum of true positives and false positives. Recall measures the model's ability to identify all relevant instances, calculated as the ratio of true positives to the sum of true positives and false negatives. To determine if a predicted object is a true positive or a false positive, an intersection over union (IoU) threshold must be determined. The IoU measures the overlap between the prediction and the true label. If the overlap of an object prediction is above the IoU threshold, then it is counted as a true positive. Otherwise, it is counted as a false positive [17].

2.3. Control System for Fruit Harvesting

2.3.1. Robot Operating System

The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. From drivers and state-of-the-art algorithms to powerful developer tools, ROS has the open-source tools needed for any robotics development [18]. One of the core advantages of ROS2 is that it is centered around modularity. ROS2 allows the robot to be broken down into subsystems that can be developed and run asynchronously.

In a typical ROS2 environment, each subsystem is organized into a node, which can be tested and run asynchronously with the rest of the system [19]. Each of these subsystems, or nodes, can communicate through topics, which are unidirectional asynchronous communication channels. Nodes can either publish to a topic (send messages to the topic) or subscribe to a topic (listen for messages on the topic). This communication can be established through publishers or subscribers created within each node. Topics can be used in one-to-one, one-to-many, many-to-one, or many-to-many relationships.

For more complicated communication, a service or an action may be used. In these communications methods, one node must host a service or action server, and other nodes may become clients of that service or action. Services and actions may create one-to-one or one-to-many relationships, as illustrated in Figure 2.

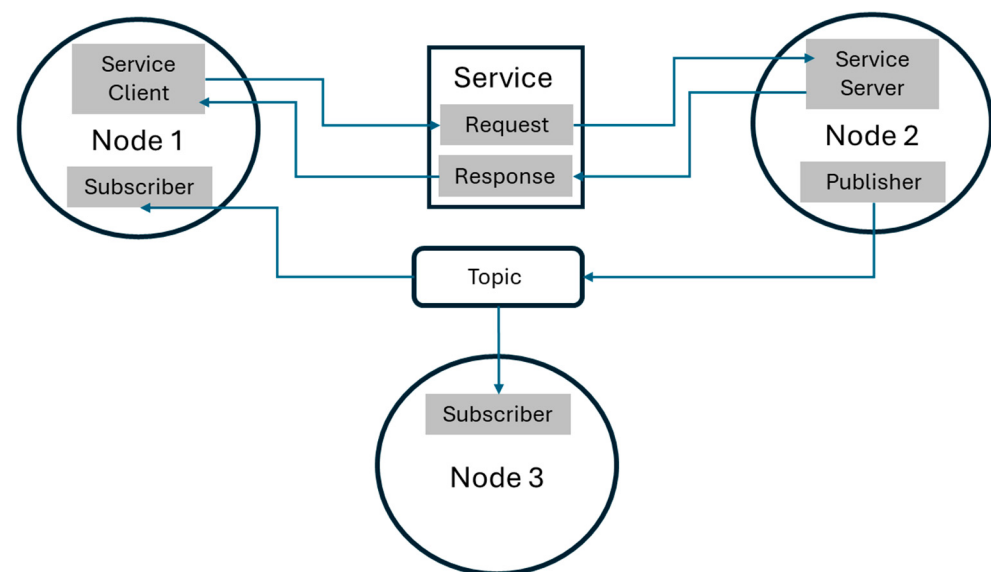


Figure 2. Example of ROS2 nodes with associated communication.

2.3.2. Designing a ROS2 Application for OrBot

The first task of the project was to redesign OrBot's software to utilize ROS2. In doing so, the primary language of the software was switched from MATLAB to Python. The first version of OrBot's control system was developed using MATLAB. While MATLAB did have some ROS2 capabilities, Python allowed much more freedom with the tools.

To integrate ROS2 into the OrBot system, subsystems needed to be identified and programmed as nodes. Major subsystems that were identified were:

1. Cameras;
2. Image processing algorithms;
3. The robotic arm;
4. The tank treads;
5. Apple picking algorithm;
6. Display;
7. Gamepad connection.

Each node was written as a class that inherited from the node class from the rclpy library. Using methods inherited from the node class, publishers and subscribers, as well as service and action servers and clients, were added in the constructor. Additionally, timers were created in the constructor for continuous repeated actions. Class methods were then created that would be called when a message was received or when a timer finished. For example, a 0.1 s timer in the camera nodes would call a method to retrieve a frame and publish it. Another example is a subscriber in the tank treads node that would call a method

to change the speed of the motors whenever a message was received on the motor speed topic. To use this class, an instance of it was created, and the spin() function was called. A basic outline of this format can be seen in Figure 3.

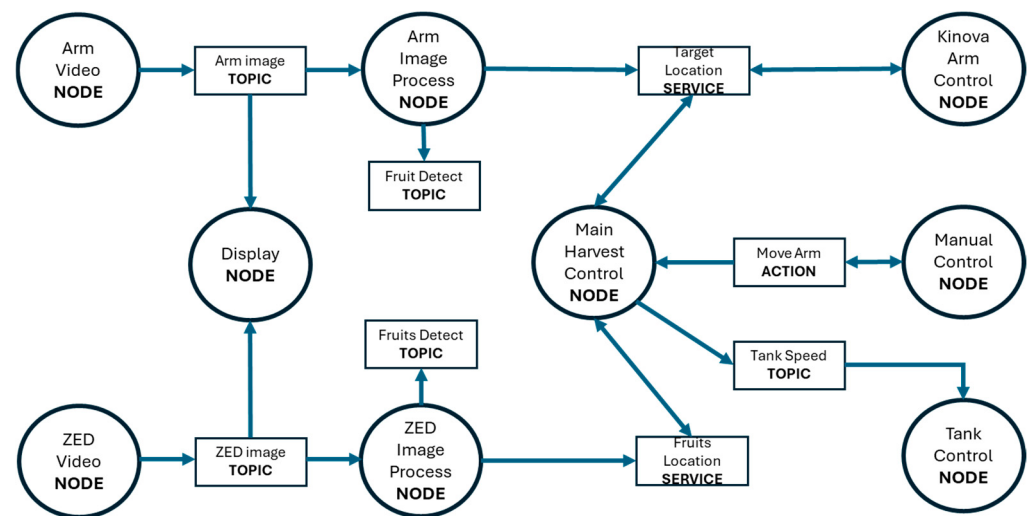


Figure 3. ROS2 nodes for fruit harvesting using OrBot.

2.4. Two-Camera System

2.4.1. Eye-in-Hand and Eye-to-Hand Systems

The next objective of the project was to add a second camera to the system with a wider field of view that would remain stationary with respect to the navigation platform. This meant that the camera needed to have eye-to-hand coordination with the robotic arm, which was a departure from the eye-in-hand coordination used by the first camera located behind the gripper.

The camera attached to the gripper was an Intel RealSense camera (RealSense, Inc., Cupertino, CA, USA) [20]. This camera and the eye-in-hand coordination used with it were excellent for performing the precise motions needed to pick apples. However, the eye-in-hand camera had a very small field of vision of 72 degrees and was unlikely to see most of the apples on a tree. A search algorithm could be implemented to have the arm move and allow the camera to scan the entire tree, but this process would be tedious and inefficient. A new camera was needed with a field of view that would encompass the entire tree.

The ZED 2, a stereo vision camera developed by StereoLabs (San Francisco, CA, USA), was utilized for this project. This camera had a much wider field of view than the eye-in-hand camera, with 120 degrees. The ZED 2 required an Nvidia graphics card to be used.

The ZED SDK allowed for the calculation of a 3D point cloud, where each pixel would be mapped to a Cartesian coordinate relative to the camera. Using the point cloud, the location of an apple identified in the image could be found with reference to the camera. This facilitated the three-dimensional location of the camera using the eye-to-hand camera.

2.4.2. Eye-to-Hand Coordination

Creating eye-to-hand coordination meant translating what the camera saw to coordinates for the robotic arm to move to. The vector from the camera to the apple can be found using the 3D point cloud. However, to move the arm in front of the apple, the vector from the base of the robotic arm to the apple needed to be known. This can be calculated using the vector from the camera to the apple, $\vec{r}_{CA}$, and the vector from the base of the robot to the eye-to-hand camera, $\vec{r}_{RC}$, as shown in Figure 4.

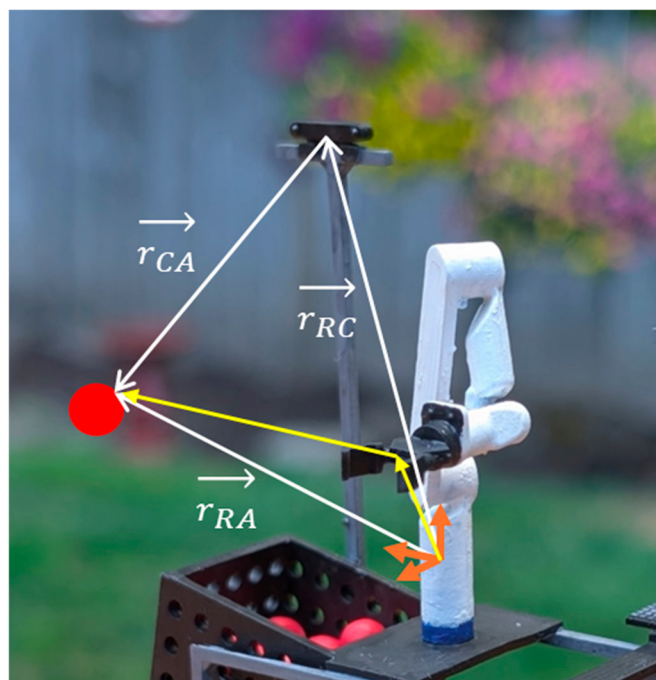


Figure 4. Position vector relationship between the camera, fruit, and manipulator.

The vector from the robot to the camera was measured physically and subtracted from the vector obtained from the point cloud. By translating the coordinates, eye-to-hand coordination was established using the following equation,

$$\vec{r}_{RC} + \vec{r}_{CA} = \vec{r}_{RA}$$

Position vector $\vec{r}_{RC}$ defines the location of the eye-to-hand camera with respect to the origin of the robot, which is defined by the extrinsic camera parameter matrix,

$$\begin{bmatrix} 1 & 0 & 0 & -0.361 \text{ m} \\ 0 & 1 & 0 & +0.795 \text{ m} \\ 0 & 0 & 1 & -0.175 \text{ m} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Position vector $\vec{r}_{CA}$ defines the position of the fruit with respect to the eye-to-hand camera. When these two position vectors are added, we get the position vector $\vec{r}_{RA}$, which was used to direct the end effector towards the apple fruit, and the precise picking movement was controlled by the eye-in-hand camera, as shown by the yellow position vectors.

2.5. Evaluation of Fruit Harvesting

The performance of OrBot for fruit harvesting was evaluated, both in the laboratory and in a commercial field (Figure 5). In the laboratory, two artificial trees with five artificial fruits randomly positioned on the trees were used for the test. Lighting in the laboratory was provided by the room's fluorescent lights. During the test, the fruit picking success rate and the picking time were measured. The picking success rate was evaluated as the ratio of the total number of fruits harvested to the total number of fruits harvestable by the robot. When the eye-to-hand camera searches for fruit, it records the fruits that are reachable, and these reachable fruits are counted as harvestable fruits. OrBot was programmed to start from the first artificial tree, harvest the fruits, and move to the second artificial tree to

harvest the remaining fruits. The fruits were then re-attached to the trees, and the test was replicated five times.

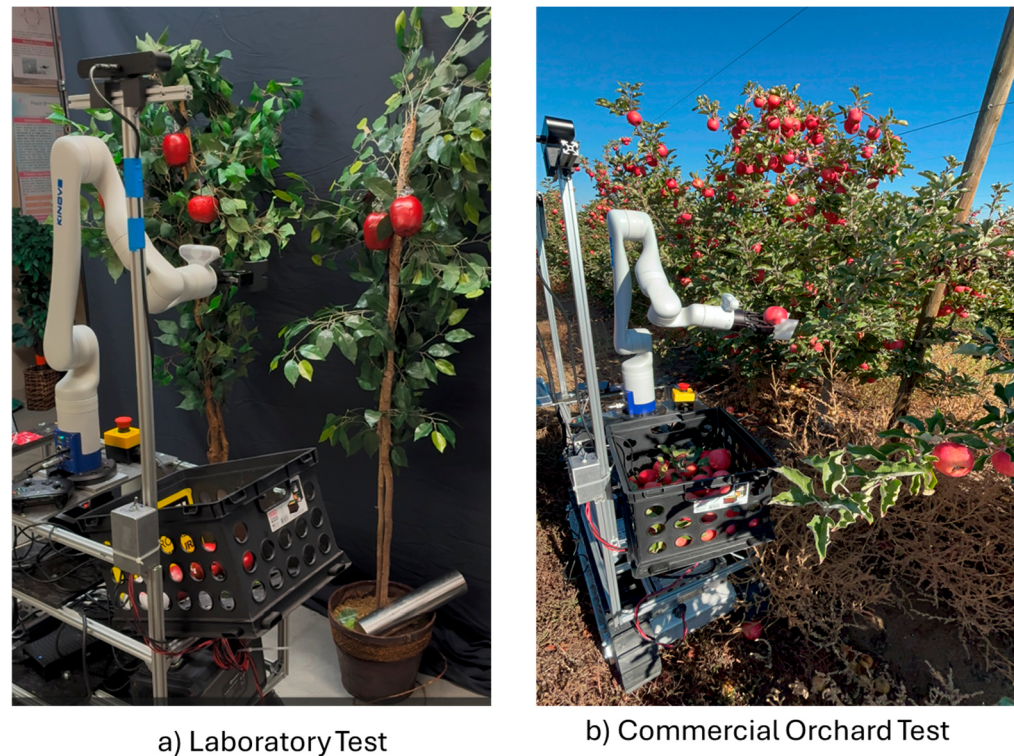


Figure 5. Evaluation of fruit harvesting with (a) laboratory test and (b) commercial orchard test.

OrBot was also tested in a commercial apple orchard, Symms Fruit Ranch, which is located in Caldwell, Idaho, USA. The apple variety in the orchard is Pink Lady apples. The apple trees, which are semi-dwarf, are trained using a spindle-type structure and were spaced about two meters apart. Twenty trees on the same row were used for the harvesting test. Similar to the laboratory test, OrBot was programmed to start from the first tree in the row. The eye-to-hand camera searched the fruits in its field of view and identified harvestable fruits. When there were no more harvestable fruits in its field of view, OrBot moved forward until a harvestable fruit was identified, and the robot stopped. The harvesting test was conducted during midday and with fine weather conditions in November 2024 and November 2025.

3. Results

3.1. Indoor Test

In all the replications of the indoor test, OrBot was able to successfully recognize and pick all the fruits. Figure 6 shows the harvesting sequence of finding the fruits, centering on the fruit, and harvesting the fruit. It can be seen from the figure the difference between the field of view of the two cameras. The camera-to-hand has a wider field of view, while the camera-in-hand has a narrower field of view.

3.2. Outdoor Test

For the outdoor test, harvesting was considered a success if the target fruit was recognized and removed from the tree. Figure 7 shows (a) the eye-to-hand camera view when searching for a target fruit and (b) the eye-in-hand camera view for precision movement to pick fruit. Figure 7c,d shows the posture of the robot during a fruit search and fruit picking. OrBot moved autonomously along the row and stopped when a target fruit was recognized.

OrBot harvested 120 apple fruits from the row of twenty trees with a success rate of 75%. In the second harvesting season, OrBot harvested 150 apple fruits from the same row of trees with a success rate of 79%. The diameters of the fruits ranged from 6 cm to 8 cm, and the fruit weights ranged from 100 g to 200 g. There were several fruits that had indentations in them that were caused by the gripper, but these were still counted as a successful harvest.

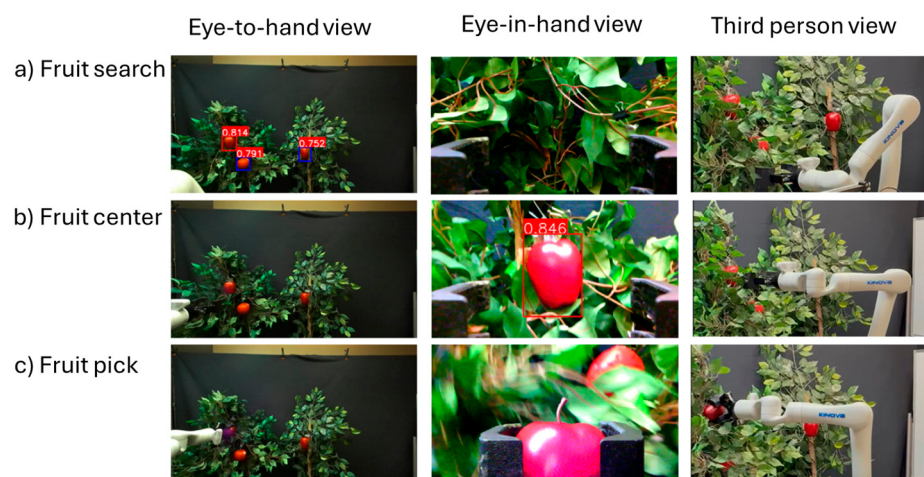


Figure 6. Fruit harvesting sequence in the laboratory test.

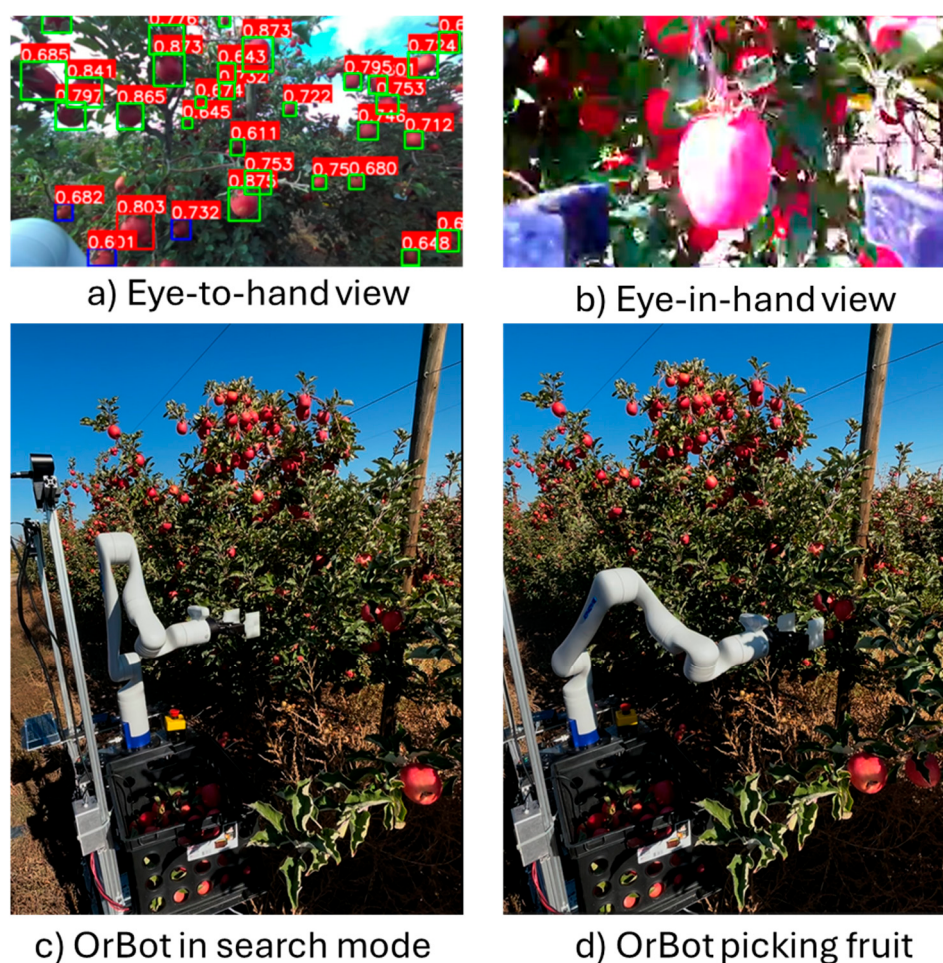


Figure 7. Fruit harvesting test in a commercial orchard showing the (a) eye-to-hand view, (b) eye-in-hand view, (c) OrBot in search mode, and (d) OrBot picking fruit.

4. Discussion

The 100% harvesting success rate of OrBot in the indoor test proves the capability of OrBot to pick fruit autonomously. In the indoor test, OrBot moved from one tree to the other without any issues. The indoor test also showed that the two-camera system facilitated searching for the fruits. The eye-to-hand vision system searched for fruits while it was moving. Once a target fruit was identified, OrBot stopped and transferred control to the eye-in-hand and conducted a more precise motion by centering on the target fruit before it approached and picked the fruit. After the fruit was picked, control was transferred again to the camera-to-hand vision system.

When moving to the commercial orchard, the harvesting success rate of OrBot decreased to about 77%. The failed attempts can be attributed to hardware and software failures. The software failures are (a) failure to recognize fruit and (b) fruit was recognized, and OrBot tried to pick another fruit (tracking failure). The hardware failures are that (c) a fruit was recognized but not picked by the end effector (end effector failure) and (d) a fruit was recognized, but the end effector was not able to approach the fruit (manipulator failure). The distribution of these failed attempts due to these factors is shown in Table 1.

Table 1. Distribution of harvesting failures.

Factors for Harvesting Failures	Percentage (%)
(a) Fruit recognition failure	16
(b) Tracking failure	17
(c) End effector failure	50
(d) Manipulator failure	17

In the failure to recognize the fruit, there were a few instances where OrBot was in front of the tree with fruit, and there was no fruit that was recognized. The main reason for this was the lighting condition. A review on fruit robotic harvesting performance listed lighting variation as one of the exogenous disturbances that affected harvesting performance [21]. When the robot and the fruit position caused a backlighting condition, the fruit appeared dark in the image. This is unlike the indoor tests, where there were no issues with fruit recognition because the fruits were always front-lit. However, when the robot was repositioned, it was able to recognize the fruit. One of the solutions that will be tried in future tests will be to include all different types of lighting conditions, such as cloudy, sunny, front light, and back light, in the training of the fruit recognition algorithm.

The other failure due to the software was when the robot harvested a different fruit from the target fruit. This was considered a failed harvesting because the target fruit was not harvested. However, after the different fruit was harvested, the robot moved towards the target fruit and harvested it. There were also instances where the robot would center between the target fruit and a nearby fruit, and it would remain in this loop. Currently, the size of the fruit in the image is used for object-tracking purposes. The relative position of the fruit with respect to the robot origin should be added for object tracking.

In the instance where the fruit was recognized but not picked by the end effector, it was an issue of the end effector design [22]. The current end effector uses a standard two-finger gripper that has customized fingers designed to capture the curvature of the apple fruit. As mentioned in the results, the harvested apple fruits had a large variation in size. The robot had difficulty with harvesting smaller fruits as the fruits slipped out of its fingers during rotation. A study for an end effector for tomatoes noted that the fruit's major diameter affected fruit grasping [23]. The end effector needs to be modified to ensure that the fruit is picked. The use of vacuum pickers [24] was also tested during this harvesting season, as

shown in Figure 8. Initial tests showed that it improved the picking efficiency to more than 90%, but further tests and design modifications are needed to validate its performance.

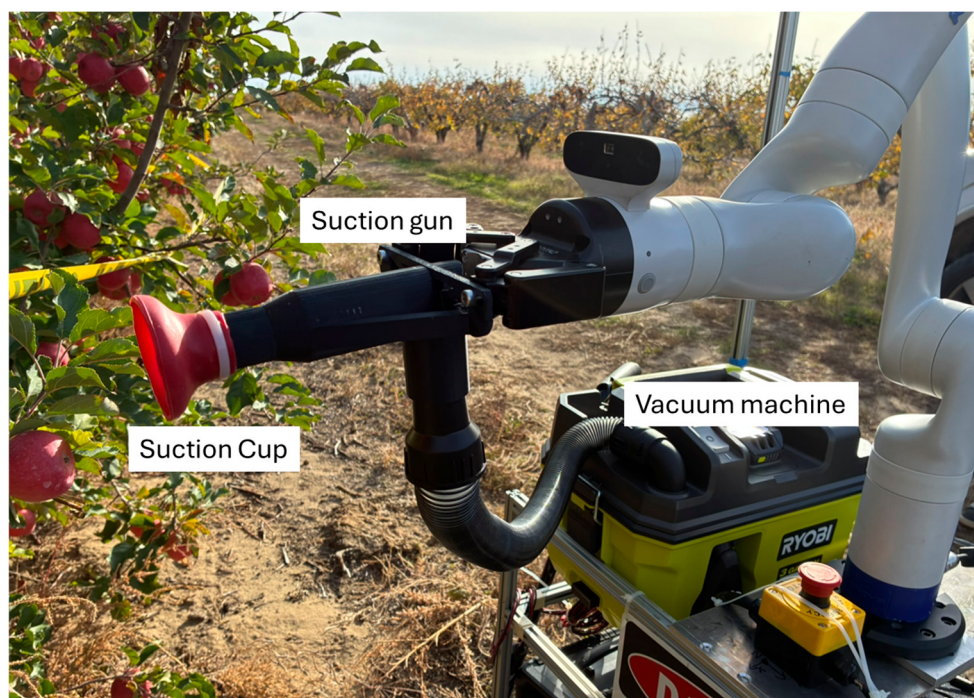


Figure 8. OrBot harvesting apples using a vacuum-based end effector.

There were also a few times where the fruits were recognized, but the manipulator did not move. The robot estimated that the fruit was outside of its work envelope. The issue was with the eye-to-hand vision system. During the search section of the eye-to-hand vision system, the control system used the cloud point data combined with the color image to estimate the distance of the fruit. If one of the cloud points on the fruit is selected for distance estimation, and if this point falls outside the work envelope, then the robot will not move. The overall area of the fruit should be used and averaged to estimate the distance of the fruit. In addition, it will be useful to combine the infrared distance sensor of the camera-in-hand to double-check the target fruit's distance.

5. Conclusions

This study presented an improved robotic fruit harvesting system, Orchard roBot (OrBot), designed to address labor shortages in specialty crop production through automation. By integrating a dual-camera vision system, transitioning the control architecture to Robot Operating System 2 (ROS2), and implementing a modular, Python-based software framework, the updated OrBot platform demonstrated enhanced fruit detection, system coordination, and harvesting performance compared to previous iterations.

Experimental evaluation in both laboratory and commercial orchard environments confirmed the effectiveness of the proposed approach. OrBot achieved a 100% harvesting success rate in controlled indoor conditions and a 75–80% success rate in outdoor orchard tests, with improved performance observed after orchard pruning. The dual-camera configuration proved particularly effective, enabling an efficient wide-area fruit search using the eye-to-hand vision system and precise manipulation using the eye-in-hand system. These results validate the feasibility of combining complementary vision modalities to overcome the limited field of view and precision constraints inherent in single-camera harvesting systems.

Field testing also revealed key challenges that must be addressed to further improve reliability and commercial viability. Variable lighting conditions, particularly backlighting, reduced fruit recognition accuracy, highlighting the need for more robust training datasets and adaptive vision algorithms. Limitations in the current gripper design affected harvesting performance for smaller fruits, suggesting that alternative end effector designs, such as vacuum-based systems, may offer improved consistency. Additionally, depth estimation and object-tracking errors in cluttered environments emphasize the importance of integrating multi-sensor fusion and improved distance estimation strategies.

Overall, the results of this study demonstrate that robotic fruit harvesting is a viable and promising solution for augmenting human labor in orchard environments. While further refinement is required to increase speed, robustness, and economic feasibility, the advancements presented in this work represent a significant step toward practical deployment of autonomous harvesting systems capable of supporting sustainable food production in the face of growing labor constraints.

Author Contributions: Conceptualization, D.M.B.; methodology, D.M.B., C.R. and C.S.; software, C.R. and C.S.; validation, C.R., C.S. and D.M.B.; formal analysis, C.R., C.S. and D.M.B.; investigation, C.R., C.S. and D.M.B.; resources, D.M.B.; data curation, C.R., C.S. and D.M.B.; writing—original draft preparation, D.M.B., C.R. and C.S.; writing—review and editing, D.M.B., C.R. and C.S. supervision, D.M.B.; project administration, D.M.B.; funding acquisition, D.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported in part by the Idaho State Department of Agriculture through the Specialty Crop Block Grant Program.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the support of Symms Fruit Ranch for allowing the use of OrBot in their commercial orchards. GenAI (Copilot for Microsoft 365) was used to check on grammar and improve sentence structure. GenAI was not used to generate any data, figures, or results.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. van Dijk, M.; Morley, T.; Rau, M.L.; Saghai, Y. A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050. *Nat. Food* **2021**, *2*, 494–501. [[CrossRef](#)] [[PubMed](#)]
2. Wei, X.; Campbell, B.L.; Khachatryan, H.; Brumfield, R.G. What Firms Hire H-2A Workers? Evidence from the US Ornamental Horticulture Industry. *HortScience* **2023**, *58*, 375–382. [[CrossRef](#)]
3. Karunathilake, E.M.B.M.; Le, A.T.; Heo, S.; Chung, Y.S.; Mansoor, S. The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture. *Agriculture* **2023**, *13*, 1593. [[CrossRef](#)]
4. Botta, A.; Cavallone, P.; Baglieri, L.; Colucci, G.; Tagliavini, L.; Quaglia, G. A Review of Robots, Perception, and Tasks in Precision Agriculture. *Appl. Mech.* **2022**, *3*, 830–854. [[CrossRef](#)]
5. Kootstra, G.; Wang, X.; Blok, P.M.; Hemming, J.; van Henten, E. Selective Harvesting Robotics: Current Research, Trends, and Future Directions. *Curr. Robot. Rep.* **2021**, *2*, 95–104. [[CrossRef](#)]
6. Hou, G.; Chen, H.; Jiang, M.; Niu, R. An Overview of the Application of Machine Vision in Recognition and Localization of Fruit and Vegetable Harvesting Robots. *Agriculture* **2023**, *13*, 1814. [[CrossRef](#)]
7. Williams, H.A.M.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [[CrossRef](#)]
8. Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. A review of key techniques of vision-based control for harvesting robot. *Comput. Electron. Agric.* **2016**, *127*, 311–323. [[CrossRef](#)]
9. Waltman, J.; Buchanan, E.; Bulanon, D.M. Nighttime Harvesting of OrBot (Orchard RoBot). *AgriEngineering* **2024**, *6*, 1266–1276. [[CrossRef](#)]

10. Bulanon, D.M.; Burr, C.; DeVlieg, M.; Braddock, T.; Allen, B. Development of a Visual Servo System for Robotic Fruit Harvesting. *AgriEngineering* **2021**, *3*, 840–852. [CrossRef]
11. Discover Our Gen3 Robotic Arm. Available online: <https://www.kinovarobotics.com/product/gen3-robots> (accessed on 10 February 2026).
12. Ruybalid, C.; Salisbury, C.; Bulanon, D.M. Robotic Fruit Harvesting with Dual Camera System. In Proceedings of the 2025 IEEE Opportunity Research Scholars Symposium (ORSS), Atlanta, GA, USA, 1 April–1 July 2025; pp. 1–4.
13. ZED 2 Versatile Stereo Camera for Spatial Perception. Available online: <https://www.stereolabs.com/products/zed-2> (accessed on 10 February 2026).
14. ZED Box Orin. Available online: <https://www.stereolabs.com/docs/embedded/zed-box-orin> (accessed on 10 February 2026).
15. HD2 Treaded ATR Tank Robot Platform. Available online: <https://www.superdroidrobots.com/store/product=789> (accessed on 1 April 2026).
16. Ultralytics YOLOv5. Available online: <https://docs.ultralytics.com/models/yolov5/> (accessed on 10 February 2026).
17. Ruybalid, C. Improving OrBot. Bachelor’s Thesis, Northwest Nazarene University, Nampa, ID, USA, April 2025.
18. ROS—Robot Operating System. Available online: <https://www.ros.org/> (accessed on 10 February 2026).
19. ROS 2 Documentation. Available online: <https://docs.ros.org/en/foxy/index.html> (accessed on 10 February 2026).
20. Intel RealSense. Available online: <https://www.intel.com/content/www/us/en/products/sku/98319/intel-realsense-depth-module-d410/specifications.html> (accessed on 10 February 2026).
21. Zhou, H.; Wang, X.; Au, W.; Kang, H.; Chen, C. Intelligent robots for fruit harvesting: Recent developments and future challenges. *Precis. Agric.* **2022**, *23*, 1856–1907. [CrossRef]
22. Silwal, A.; Davidson, J.R.; Karkee, M.; Mo, C.; Zhang, Q.; Lewis, K. Design, integration, and field evaluation of a robotic apple harvester. *J. Field Robot.* **2017**, *34*, 1140–1159. [CrossRef]
23. Li, Z.; Miao, F.; Yang, Z.; Chai, P.; Yang, S. Factors affecting human hand grasp type in tomato fruit-picking: A statistical investigation for ergonomic development of harvesting robot. *Comput. Electron. Agric.* **2019**, *157*, 90–99. [CrossRef]
24. You, K. Development of an adaptable vacuum based orange picking end effector. *Agric. Eng. Int. CIGR J.* **2019**, *21*, 58–66.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.