AI-based Fruit Harvesting using a Robotic Arm

Hovannes Kulhandjian[1], Nicholas Amely[1], and Michel Kulhandjian[2]

[1]Department of Electrical and Computer Engineering, California State University, Fresno, CA, USA.

[2]Department of Electrical and Computer Engineering, Rice University, TX, USA.

Emails: hkulhandjian@csufresno.edu, namely@mail.fresnostate.edu, michel.kulhandjian@rice.edu

A paper from the Proceedings of the
16th International Conference on Precision Agriculture
21-24 July 2024
Manhattan, Kansas, United States

Abstract.

This paper presents an innovative AI-based fruit harvesting system utilizing a robotic arm, designed to enhance efficiency and reduce labor intensity in agricultural practices. Integrated into the Smart Agricultural Robot, Bulldog (SARDOG), the system leverages a six-degrees-of-freedom robotic arm controlled by a Raspberry Pi 5 and employs an NVIDIA Jetson Orin for advanced fruit detection and localization using a YOLOv8 deep learning model. The system facilitates real-time fruit detection and harvesting through seamless hardware communication and precise kinematic control. Extensive dataset compilation and training were conducted to ensure high accuracy in fruit classification and localization. Experimental results demonstrate the system's capability to accurately detect and pick various types of tree fruits, optimizing the harvesting process. This integration into the SARDOG provides a stable and efficient platform for agricultural automation, representing a significant advancement in precision agriculture. The research underscores the potential for AI and robotics to revolutionize farming, meeting the increasing global demand for food while promoting sustainability.

Keywords.
Precision Agriculture, Robotic Arm, Fruit Detection, Deep Learning, Autonomous Harvesting.

I. Introduction

In recent years, precision agriculture has become crucial to the farming industry, revolutionizing traditional practices. By incorporating advanced technologies like GPS, IoT sensors, drones, robotic arms, and data analytics, precision agriculture enables farmers to optimize resource utilization, increase crop yields, and reduce environmental impact [1-4]. These innovations provide real-time monitoring and management of crops, soil, and weather conditions, leading to more informed decisions and efficient farm operations. Additionally, precision agriculture supports sustainable practices by minimizing waste, reducing the use of pesticides and fertilizers, and conserving water. As the global demand for food continues to rise, embracing precision agriculture is key to meeting this demand while ensuring the sustainability of farming.

Harvesting is arguably the most fundamental task in agriculture, with its efficiency playing a crucial role in the success of farms. This task is often the most labor-intensive during the growing season, particularly on smaller farms with limited equipment. To address inhumane labor conditions and the increasing demand for agricultural products, many autonomous robotic harvesting systems have been developed. This paper provides a comprehensive analysis of current robotic harvesting solutions and introduces a newly proposed system with unique advantages.

We present a smart harvesting system that employs a standard six-degrees-of-freedom robotic arm to detect and pick eight different kinds of tree fruit. This system is installed on our custom Smart Agricultural Robot, Bulldog (SARDOG) [1]. The system utilizes a Raspberry Pi 5 single-board computer to control the robotic arm's movements and an NVIDIA Jetson Orin Artificial Intelligence (AI) computer to optimize fruit detection. This setup replaces the previously used Raspberry Pi 4B computer, enabling faster and more accurate results. The Raspberry Pi 5 runs a series of connected Python scripts for movement control, while the Ultralytics YOLOv8 (You Only Look Once) computer vision deep learning model processes the RGB USB camera's real-time feed on the Jetson Orin. Hardware communication between the two onboard computers ensures seamless integration of detection and harvesting components, converting virtual positional data to real-world positions through a pixel coordinates-to-joint control system. Our system uses an electronic adaptive gripper to grasp the fruits, which differs from other systems that use suction or cutting mechanisms to harvest the fruits [5]. Since our gripper is fully electronic, a large air compressor is not needed which saves power and makes our system more efficient. The advantage of the simplicity of our gripper as opposed to a cutting mechanism is that ours is more reliable and requires less maintenance, in addition to producing harvested fruits without partial stems still attached.

Training the machine learning model involved a trial-and-error approach using comprehensive fruit datasets compiled into a master dataset with Roboflow. Custom compiling the master dataset enabled class name changes and bounding box annotation corrections to be made, which enabled the model to be more efficient in our specific system. Other systems, such as the one by Salim *et al.* [7], use pre-compiled master datasets like Fruit-360 to train their deep-learning models. This approach can lead to longer training times due to unnecessary classes and poorer results due to limited customization. In contrast, our model was trained and fine-tuned on Google Colab, running multiple trials with modified parameters until achieving sufficient accuracy.

This paper culminates recent research and development efforts, detailing the current smart harvesting system's development and its implications for improving labor conditions and efficiency in agriculture. The following sections discuss the system's development, testing, and performance.

The rest of the paper is organized as follows. In Section II, we discuss the robotic arm movement system development. In Section III, we discuss the detection system development including the machine learning algorithm training. In Section IV, we present the SARDOG integration and experimentation before presenting our conclusion in Section V.

II. Robotic Arm Movement System Development

A. Hardware System Overview

The following hardware components make up the smart harvesting system:

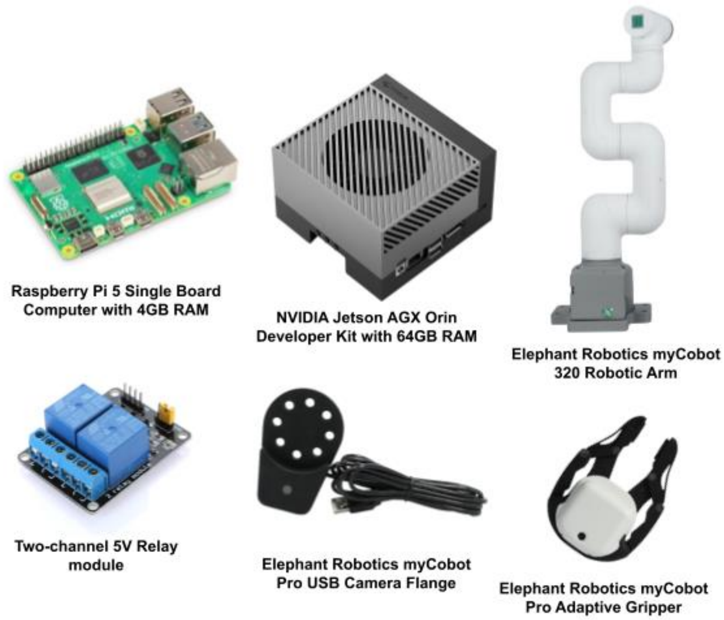


Fig. 1. System Hardware.

The most crucial hardware components of the smart harvesting system are the Jetson Orin computer and the robotic arm.

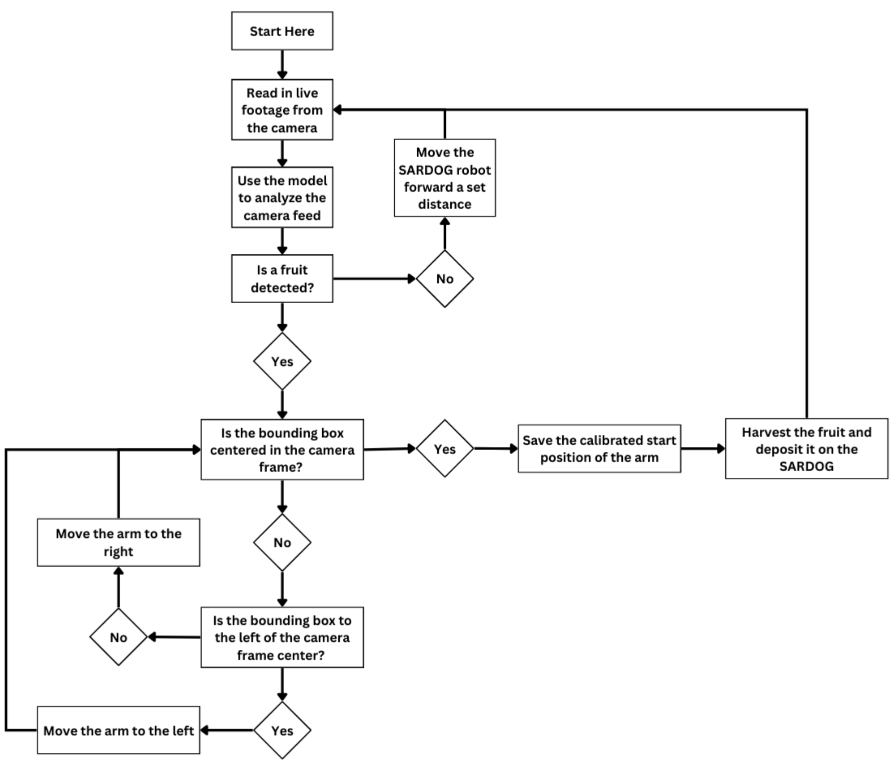


Fig. 2. System Behavior Flowchart.

The flowchart in Figure 2 details the system's operation from start to finish. The process begins by analyzing real-time camera footage using the YOLOv8 deep learning model. When fruit is detected, the model's bounding box coordinates are sent to the pixel coordinate processing Python script on the Jetson Orin. This script determines whether the arm is already centered on the fruit or needs to rotate right or left to become centered. If the arm is not centered, it rotates its base joint as instructed by the Jetson Orin. Once centered, the arm saves its base joint position and harvests the fruit. The harvested fruit is then stored in a bin on the SARDOG, and the robot moves forward until another fruit is detected.

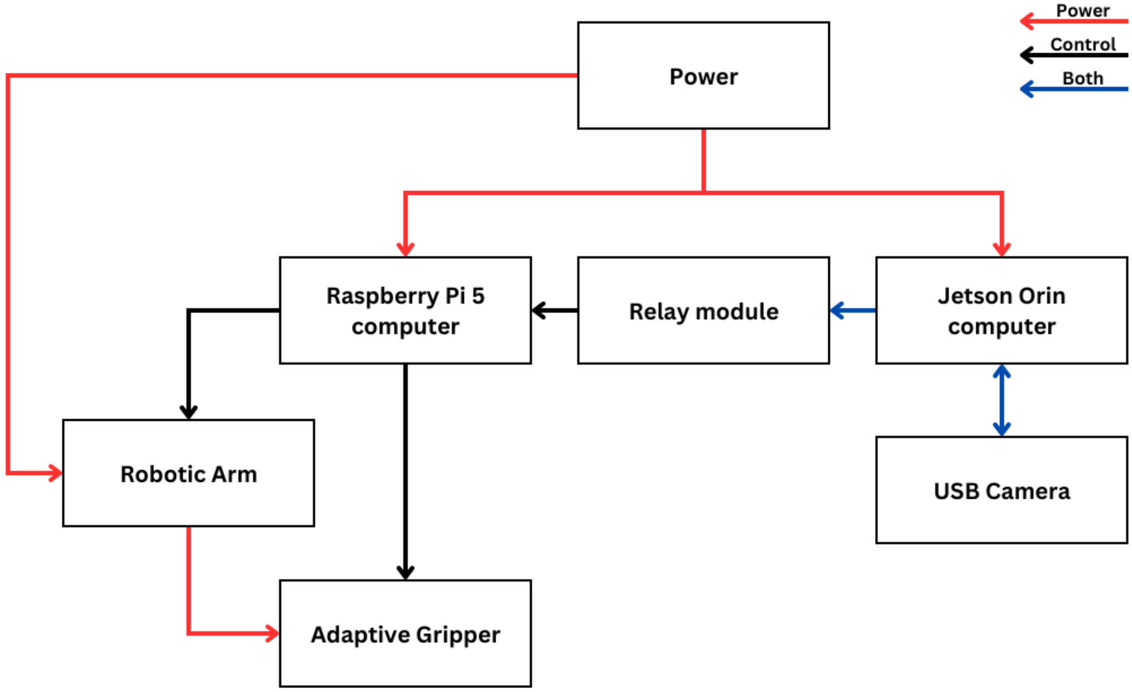


Fig. 3. Generic Hardware Block Diagram.

Figure 3 illustrates the block diagram detailing the basic hardware flow of the smart harvesting system. The two onboard computers, the Raspberry Pi 5 and the Jetson Orin communicate via two 5V relays that connect their GPIO headers. The Jetson Orin powers the USB camera, capturing real-time video footage, which is processed by the AI model. The Jetson Orin then sends directional and verification Boolean signals to the relay modules.

The directional command relay sends a HIGH value to indicate that the robotic arm should adjust its starting position to the right, and a LOW value to adjust to the left. The verification command relay sends a HIGH value to signal that the arm is centered on the detected fruit and does not need further adjustment; a LOW value indicates that the arm should continue reading input from the directional command relay.

Relays are used instead of direct GPIO connections because the Raspberry Pi's header is connected to an Elephant Robotics hat board within the arm, operating at 24V, while the Jetson Orin's GPIO header operates at 5V. Once the Raspberry Pi confirms the arm is centered on the fruit, it sends angular values to the servos in the robotic arm and adaptive gripper.

The system is powered by the SARDOG’s power distribution board, connected to its two 24V batteries. This board provides 24V to the robotic arm and gripper and steps down the voltage to supply 5V to both onboard computers through 24V, 12V, and 5V converters.

B. Software System Overview

The Raspberry Pi 5 operates using the latest Debian Raspberry Pi Desktop Operating System and controls the servos in the robotic arm via a hat board within the arm. We developed two main Python scripts to handle the kinematics necessary for the harvesting process.

The first script, the analysis script, moves the arm back to its maximum extent and then pans it horizontally. As the arm moves laterally, it reads input from the relays controlled by the Jetson Orin. Based on the Boolean value of the directional relay, the arm adjusts its position right or left to center itself on the detected fruit. Once centered, the second relay outputs a HIGH value, and the script saves the calibrated start position of the arm.

The second script controls the picking motion of the arm. It includes several complex motions tailored for different harvesting scenarios. These motions are programmed using angular values sent to the six servos within the arm, enabling precise and efficient fruit picking.

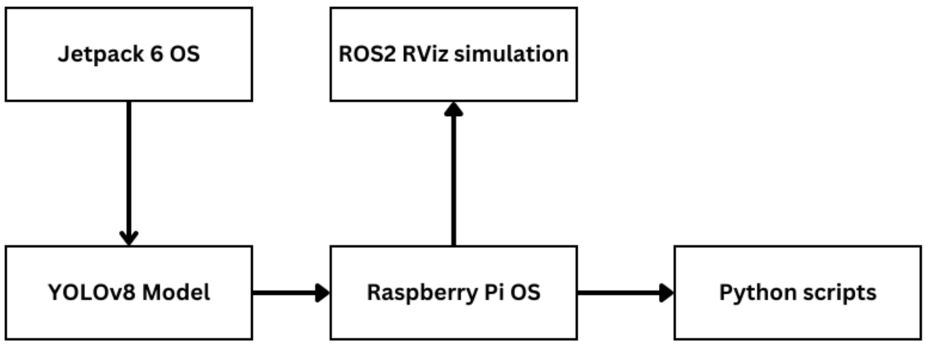
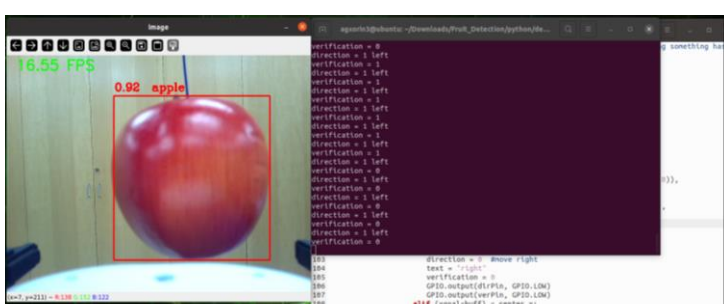


Fig. 4. Generic Software Block Diagram.

The NVIDIA Jetson Orin operates on the Ubuntu-based Jetpack 6 Operating System and handles all image processing for the smart harvesting system. Utilizing the trained YOLOv8 model and the OpenCV Python library, the system detects, classifies, and locates fruits in real-time USB camera video footage by drawing bounding boxes around them. The bounding boxes are generated using pixel coordinate data from the YOLOv8 model.

In our harvesting Python script, the center of each bounding box is determined, as illustrated in Figure 5. This center point is used to send directional and verification Boolean values to the relays. If the center of the bounding box is to the left or right of the camera frame's center, the Jetson Orin sends a corresponding Boolean value to the relays. When the center point aligns with the camera frame's center, a HIGH value is sent to the verification relay, indicating that the arm is correctly positioned for harvesting.



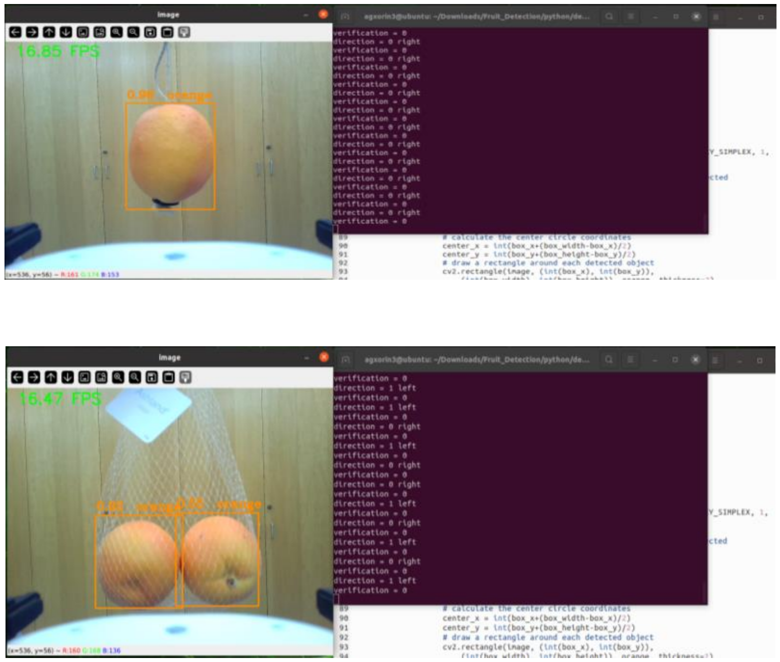
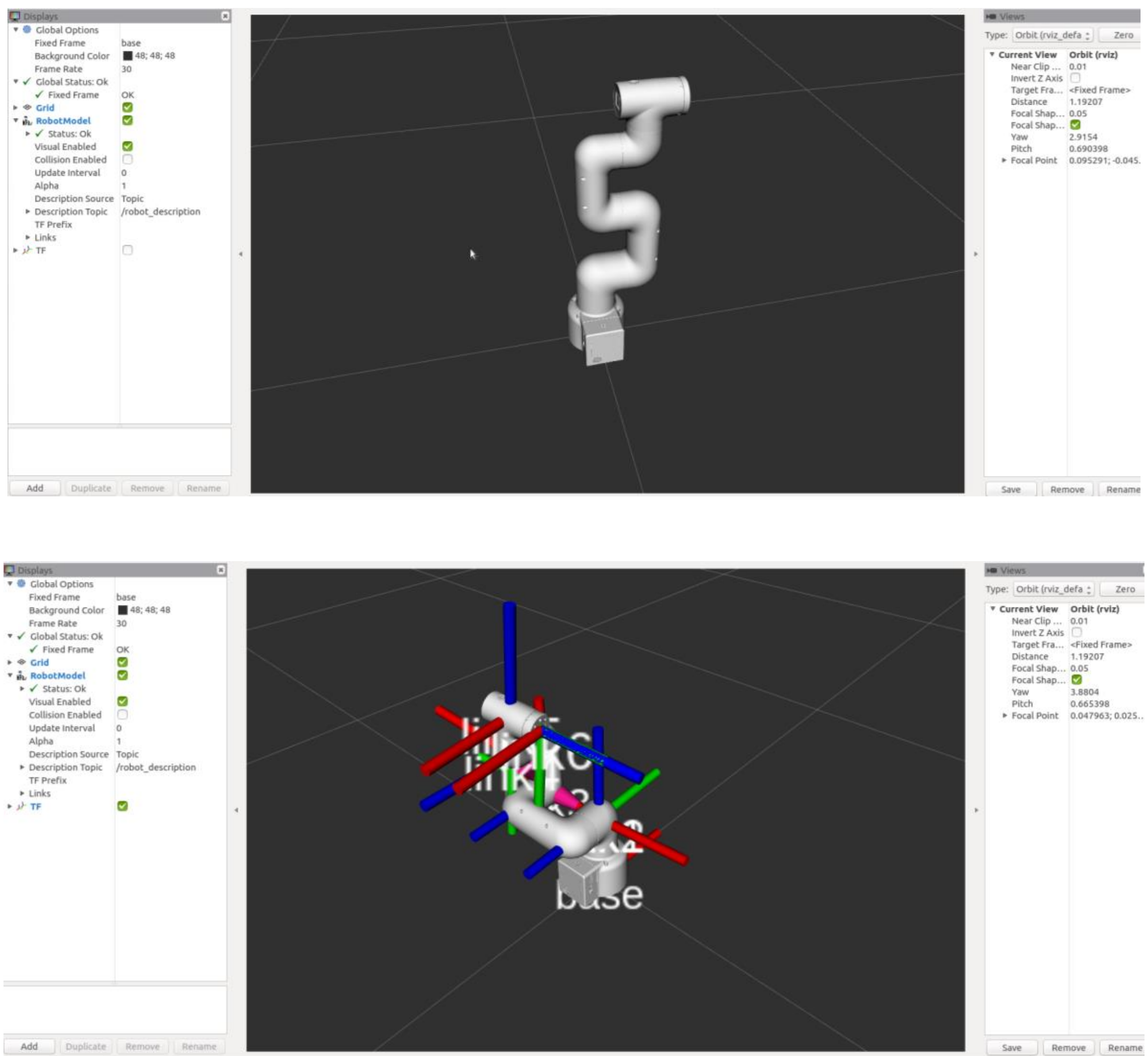


Fig. 5. Drawing bounding boxes around detected fruits.



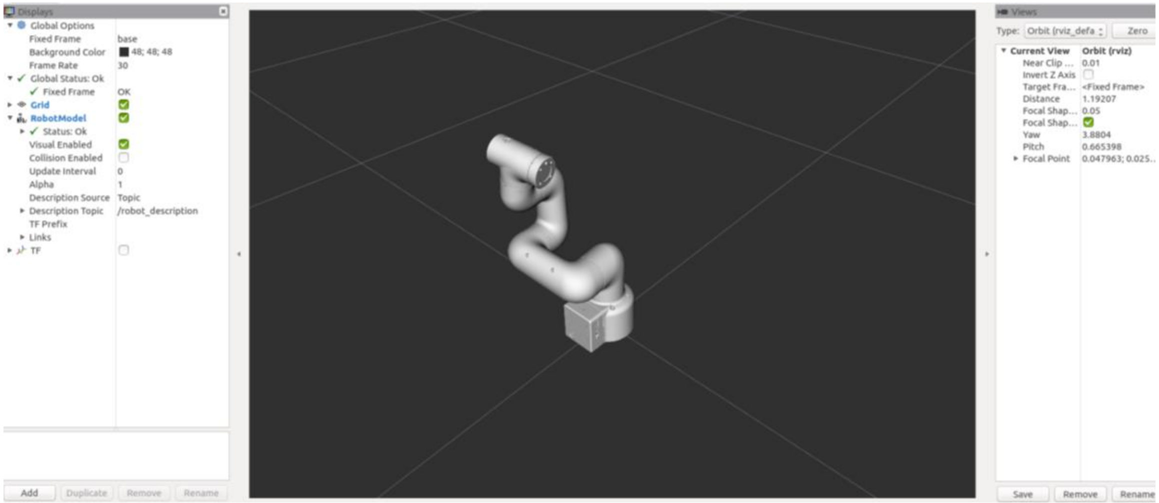


Fig. 6. ROS2 RViz Joint Control Simulation Environment.

During the development of our programmed harvesting motions for the robotic arm, virtual simulation of the joints was crucial for planning the kinematics and ensuring the mounted camera remained steady at different points. To achieve smooth and effective motions, the six servos had to move at varying speeds to reach their final angles simultaneously. We used ROS2 Foxy to simulate these motions. To optimize the harvesting motions, we employed an RViz simulation of the six joints of the robotic arm, developed by Elephant Robotics. This simulation environment is shown in Figure 6.

Due to the use of this simulation environment and the developed horizontal position calibration system, the need for coordinate control of the robotic arm was eliminated. Coordinate control of the joints is more complicated and requires many more computations on the backend since it converts a single coordinate input into motions for all of the joints of the arm. In the case of fruit harvesting, it is helpful for the arm to keep its end effector cleared of branches and leaves as it is reaching for the fruit, to reduce the risk of it getting stuck in the crop or damaging other fruits. Relative angular control, which is what we implemented in our system, enables full control of every stage of the robotic arm movements, which allows the end effector to be carefully guided around obstacles as it reaches for the fruit. Other systems that use coordinate control rely on their scripts to determine the most optimal joint movements to reach the final coordinate point, which makes the movements less customizable and gentle [6].

III. Detection System Development

A. Dataset Compilation and Formatting

The master dataset was custom-compiled using existing unlabeled image datasets from Kaggle. The eight types of tree fruits included in this dataset are shown in Figure 7. To create a dataset compatible with training a YOLOv8 model, each fruit in the images had to be labeled with bounding boxes and class names. This was accomplished using the Roboflow image annotation tool. We first divided the images into eight separate datasets, annotated all the images, and then merged them to create a final master dataset with eight classes. All images were resized to 640x640 and their orientations were normalized. The master dataset contained a total of 4,502 images of tree fruits, with 80% allocated for training and the remaining 20% for validation and testing.

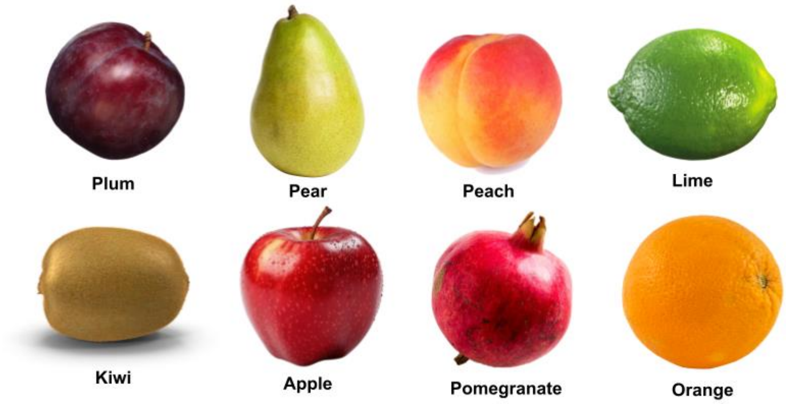


Fig. 7. Fruits in the Master Dataset.

B. YOLOv8 Computer Vision Training

YOLOv8 was selected as the computer vision model for this task due to its superior object localization capabilities compared to traditional models like ResNet50 and GoogleNet. This precision is essential for the smart harvesting system, as the robotic arm must accurately center itself on the detected fruit, which requires precise pixel coordinate information in the image frame. Training for our model was conducted using Google Colab, leveraging its accelerated GPU performance. Specifically, an NVIDIA T4 Tensor Core GPU was utilized for training. Initially, the model was run for only 8 epochs, which did not yield consistent accuracy. Although it increased the training time significantly, the final model was trained for 25 epochs, resulting in much better performance.

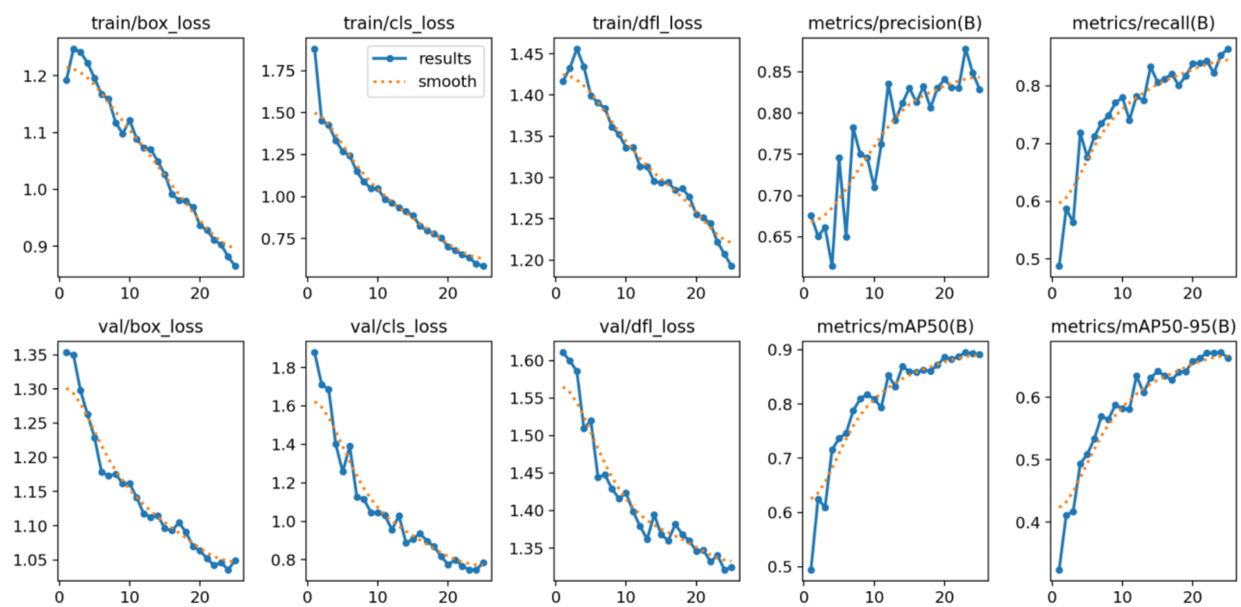


Fig. 8. Training and Validation Results.

The results, shown in Figure 8, demonstrate the model's performance during training on our custom dataset. The training process took a total of 56 minutes, averaging about 2 minutes per epoch. After 20 epochs, the model's predictions stabilized at an 88% validation accuracy. These results indicate that the model can differentiate between eight different types of tree fruits with sufficient accuracy. The primary objective, however, was to detect and localize tree fruits in general, and these results confirm that the model can accomplish this task with high accuracy.

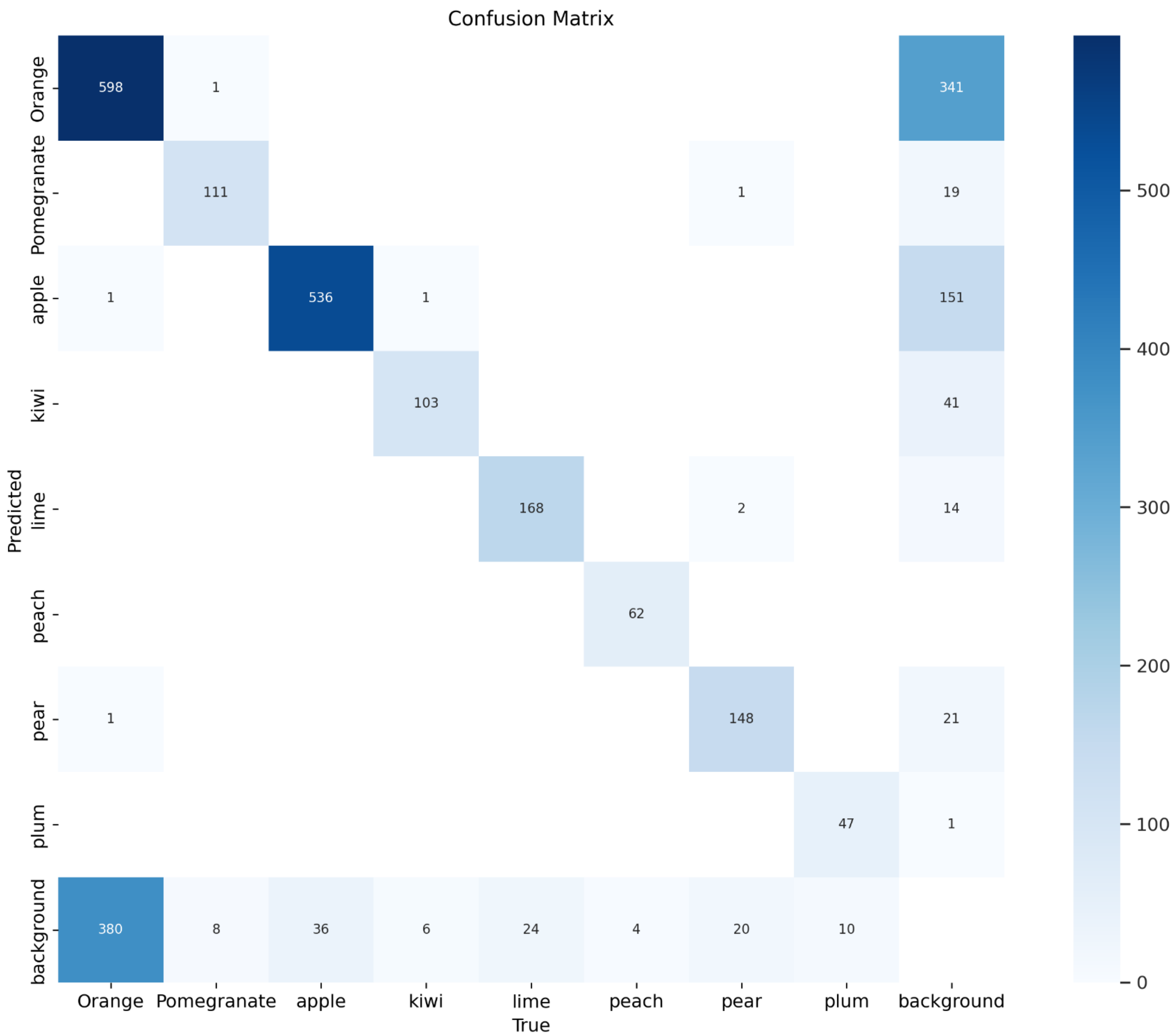


Fig. 9. Confusion Matrix.

The confusion matrix above visually represents the classification predictions made by the deep learning model. It's evident that the model achieved significantly higher accuracy in predicting the orange and apple classes compared to other types of fruits. This aligns with the fact that the master dataset contained more images of apples and oranges, providing the model with more data to learn from for these classes.

To create a model capable of accurately classifying these eight tree fruits individually, a larger and more balanced dataset, along with a longer training time, would be necessary. However, for the sole purpose of detecting the eight classes as fruits in general, this model performs more than sufficiently.

Fig. 10. Validation Batch Example.

The collage above presents a snapshot of the validation results obtained from the deep learning model. Various image perspectives were included in the master dataset, ranging from standalone individual fruits to fruits in groups and clusters, as well as fruits hanging from trees and partially obscured by branches. The results indicate that when the fruit is clearly visible, the model classifies it with a confidence rating of 90%. However, when the fruit is partially obscured, the confidence rating remains between 60-80%.

IV. SARDOG Integration and Testing

A. Integration

Several modifications were implemented to seamlessly integrate the smart harvesting system into the SARDOG frame. The base of the robotic arm, being relatively narrow, posed challenges for affixing it to a metal rod. To overcome this limitation, a wooden base was designed to securely attach the robotic arm with four long bolts, providing ample space for accommodating other hardware components.

Once all components were mounted on the platform, four U-bolts were utilized to firmly attach the platform to the left side outermost rod of the robot. This setup allowed for easy attachment and detachment of the system, which proved essential for maneuvering the SARDOG through narrow passageways during transportation.

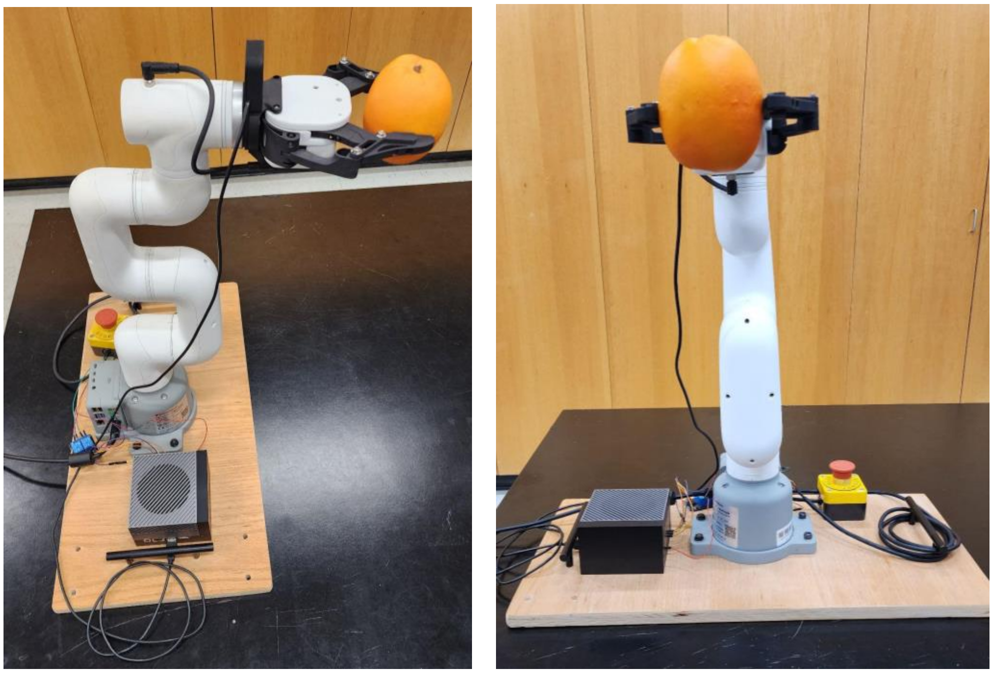


Fig. 11. Smart Harvesting System by itself.

B. Testing and Experimentations

As depicted in Figure 12, positioning the robotic arm sufficiently away from the side of the robot enabled easy access to hanging tree fruits. We observed that widening the SARDOG frame to match the length of the orchard path facilitated overall system access. This adjustment was necessary because positioning the robot closer to the left side of the path was impractical due to other onboard systems requiring it to be centered on the path.

The system demonstrated optimal performance when targeting fruits positioned on the outer edges of the trees but faced challenges reaching fruits located closer to the inner parts of the trees. This issue can be addressed by incorporating a longer arm. Despite this, the YOLOv8 model successfully detected and classified fruits positioned deeper within the trees. The arm's pulling and twisting functionalities enabled it to harvest oranges and plums without causing damage, as oranges require twisting motions while plums require pulling motions.

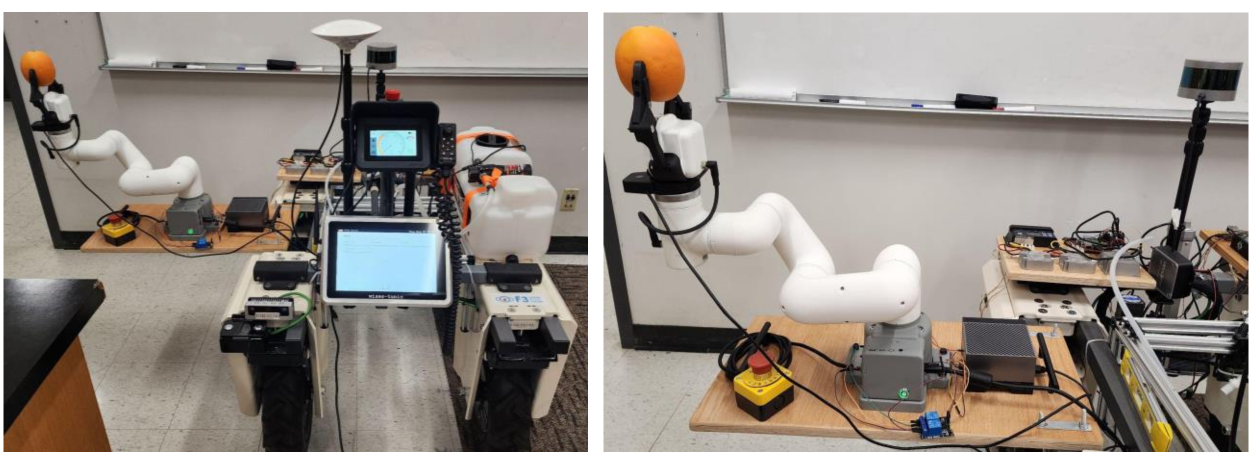


Fig. 12. Smart Harvesting System Integrated onto the SARDOG.

V. Conclusion

Through this research, we've developed an autonomous smart harvesting system for SARDOG, designed to optimize the harvesting of tree fruits. Featuring a six-degrees-of-freedom robotic arm, this system reduces the need for human labor in this labor-intensive task. Utilizing a YOLOv8 deep learning model and two onboard computers, the system achieves fruit detection and localization. This information is then translated into real-world actions through a hardware communication system and coordinate conversion mechanism.

The development of this system involved extensive research and design efforts, resulting in a solution that enhances the efficiency of one of agriculture's most challenging tasks. By deploying it on SARDOG, the system can effectively approach trees and provide a stable base for storing harvested fruit, thereby creating a highly effective tool for farmers.

Acknowledgments

We want to give a special thanks to the Economic Development Administration (EDA) Project Grant 077907908, Fresno-Merced Future of Food Innovation (F3) Coalition.

VI. References

[1] H. Kulhandjian, Y. Yang and N. Amely, "Design and Implementation of a Smart Agricultural Robot bullDOG (SARDOG)," in Proc. of IEEE International Conference on Computing, Networking and Communications (ICNC), Big Island, Hawaii, Feb. 2024.

[2] H. Kulhandjian, D. Rocha, B. Bennett, N. Amely, M. Kulhandjian, “AI-based Precision Weed Detection and Elimination,” in Proc. of 16th International Conference on Precision Agriculture, Manhattan, Kansas, July. 2024.

[3] H. Kulhandjian, B. Irineo, J. Sales, M. Kulhandjian, “Low-Cost Tree Health Categorization and Localization Using Drones and Machine Learning,” in Proc. of IEEE International Conference on Computing, Networking and Communications (ICNC), Big Island, Hawaii, Feb. 2024.

[4] H. Kulhandjian, N. Amely, M. Kulhandjian, “AI-based Pollinator using XY-Core Robot,” in Proc. of 16th International Conference on Precision Agriculture, Manhattan, Kansas, July. 2024.

[5] E. Vrochidou, V. N.Tsakalidou, I. Kalathas, T. Gkrimpizis, T. Pachidis, V. G. Kaburlasos, “An Overview of End Effectors in Agricultural Robotic Harvesting Systems”. Agriculture 2022, 12, 1240. https://doi.org/10.3390/agriculture12081240

[6] T. Yoshida, Y. Onishi, T. Kawahara, et. al., “Automated harvesting by a dual-arm fruit harvesting robot”. *Robomech Journal* 9:19, 2022. https://doi.org/10.1186/s40648-022-00233-9

[7] F. Salim, F. Saeed, S. Basurra, S.N. Qasem, T. Al-Hadhrami, “DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition,” *Electronics* 2023, *12*, 3132. https://doi.org/10.3390/electronics12143132