AI-Powered Fruit Harvesting System Using a Robotic Arm for Precision Agriculture

Hovannes Kulhandjian†, Nicholas Amely†, and Michel Kulhandjian‡

†Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740, U.S.A.
E-mail: {hkulhandjian, namely}@mail.fresnostate.edu
‡Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005, U.S.A.
E-mail: mkk6@buffalo.edu

***Abstract*—This paper introduces an AI-based fruit harvesting system designed to improve efficiency and reduce labor demands in agriculture. Integrated with the Smart Agricultural Robot Bulldog (SARDOG), the system utilizes a six-degrees-of-freedom robotic arm controlled by a Raspberry Pi 5 and an NVIDIA Jetson Orin for advanced fruit detection and localization. The YOLOv8 deep learning model facilitates real-time identification and harvesting of fruits through precise kinematic control and seamless hardware communication. A robust dataset was compiled and extensively trained to ensure high accuracy in fruit classification and positioning. Experimental results confirm the system's ability to accurately detect and harvest various types of tree fruits, significantly optimizing the fruit-picking process. This integration into SARDOG provides a stable, efficient platform for agricultural automation, marking a significant advancement in precision farming. The research highlights the transformative potential of AI and robotics in addressing global food demands while promoting sustainable agricultural practices.**

***Index Terms*—Autonomous Harvesting, Precision Agriculture, Fruit Detection and Localization, Robotic Arm, Machine Learning in Agriculture.**

I. Introduction

In recent years, precision agriculture has become an essential component of modern farming, transforming traditional agricultural practices. By incorporating advanced technologies such as GPS, IoT sensors, drones, robotic arms, and data analytics, precision agriculture allows farmers to optimize resource utilization, increase crop yields, and minimize environmental impact [1]. These technologies enable real-time monitoring and management of crops, soil, and weather conditions, leading to better decision-making and more efficient farm operations. Furthermore, precision agriculture promotes sustainable farming practices by reducing waste, decreasing pesticide and fertilizer use, and conserving water. As global food demand continues to rise, precision agriculture is increasingly necessary for meeting these demands while maintaining long-term agricultural sustainability.

Harvesting is a critical task in agriculture, often labor-intensive and time-consuming, especially for smaller farms with limited equipment. Addressing the challenges of labor shortages and the increasing demand for agricultural products, various autonomous robotic harvesting systems have been developed.

The development of autonomous robotic systems for fruit harvesting in recent years has gained momentum as a response to the growing need for efficient, labor-reducing technologies in agriculture. The study by Yoshida *et al.* [2] presents a dual-arm fruit harvesting robot designed to operate on V-shaped trellises specifically cultivated and adjusted for the robot's functionality. Utilizing sensors and computer vision, the system estimates the position of the fruit and employs inverse kinematics with a path-planning algorithm to harvest fruits like pears and apples without causing collisions with other fruits or the robot's arms. While this system demonstrates significant advancements in automation, it is largely dependent on pre-adjusted orchard structures, which may limit its adaptability to varying tree shapes and orchard conditions.

Our system, by contrast, introduces a single robotic arm mounted on the Smart Agricultural Robot Bulldog (SARDOG), designed to harvest fruits from trees in natural, unmodified orchard environments. Unlike the Smart Image Recognition Mechanism for Crop Harvesting System by Horng *et al.* [3], which also relies on a customized cultivation setup, our system employs a YOLOv8 deep learning model for real-time fruit detection and localization, combined with an adaptive electronic gripper for versatile fruit picking. This eliminates the need for suction-based or cutting mechanisms, commonly seen in the low-cost robotic arms designed by Megalingam *et al.* [4], thereby offering a lightweight and more energy-efficient alternative.

Our model's adaptability surpasses the gesture-controlled robotic arm from Yu *et al.* [5] by utilizing a broader, robust dataset tailored through iterative custom training on Roboflow, enhancing both accuracy and flexibility across various orchard layouts. This custom dataset approach contrasts with Zadeh *et al.*'s [6] reliance on pre-compiled datasets, which can lack relevancy to specific harvesting contexts. In addition, the integration of Raspberry Pi 5 and NVIDIA Jetson Orin computers enables high-speed, precise decision-making, reinforcing our system's flexibility and scalability across diverse agricultural settings.

Our system employs an electronic adaptive gripper to securely grasp fruits, offering a distinct advantage over systems that rely on suction or cutting mechanisms for harvesting such as the one discussed in [7]. Since our gripper is fully electronic, it eliminates the need for a bulky air compressor, which reduces power consumption and enhances overall system efficiency. The simplicity of our gripper design also improves reliability, requiring less maintenance compared to cutting mechanisms, and produces harvested fruits without the issue of partial stems still attached.

To train the machine learning model, we adopted an

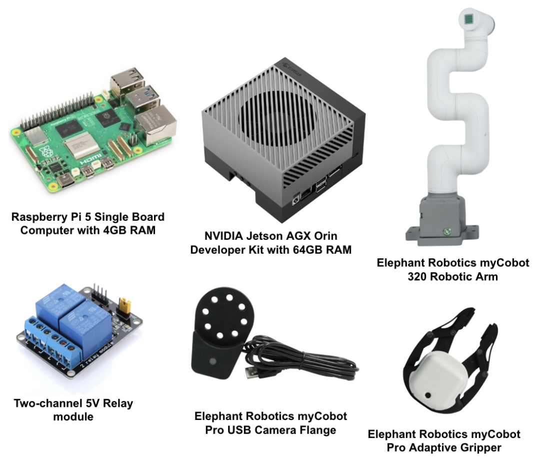

Fig. 1: System hardware overview.

iterative, trial-and-error approach using comprehensive fruit datasets, which we compiled into a custom master dataset using Roboflow [8]. This allowed us to adjust class names and refine bounding box annotations, resulting in a more efficient model tailored to our system's specific needs. In contrast to other systems, such as the one by Salim *et al.* [9], which rely on pre-compiled datasets like Fruit-360 [10], our custom dataset streamlined the training process by eliminating unnecessary classes and improving model accuracy. We conducted multiple trials and fine-tuned the model on Google Colab, adjusting parameters until we achieved the desired level of accuracy.

This paper presents an innovative AI-powered robotic system for fruit harvesting. The system uses a six-degrees-of-freedom robotic arm mounted on the custom-built SARDOG, which was first presented at the IEEE ICNC 2023 conference [11]. This paper will describe the architecture, detection, and harvesting mechanisms of the smart fruit-picking robot, along with training and experimental results.

II. Robotic Arm Movement System Development

A. Hardware System Overview

The smart fruit-harvesting system integrates the following hardware components as shown in Fig. 1.

- NVIDIA Jetson Orin AI computer
- Raspberry Pi 5 single-board computer
- Six-degrees-of-freedom robotic arm
- Adaptive electronic gripper
- USB RGB camera
- SARDOG platform with the power distribution system

The Jetson Orin handles fruit detection and processing tasks, while the Raspberry Pi 5 controls the movement of the robotic arm. The Raspberry Pi 5 and Jetson Orin communicate through GPIO relays, enabling real-time synchronization between fruit detection and picking operations.

The flowchart in Fig. 2 illustrates the system's operational workflow from initialization to completion. The process begins with the analysis of real-time camera footage using the YOLOv8 deep learning model. When a fruit is detected, the model generates bounding box coordinates, which are then

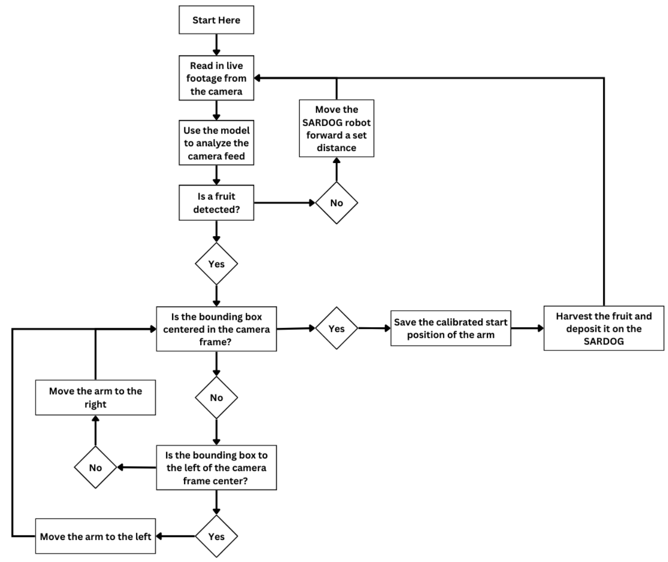

Fig. 2: System behavior flowchart.

passed to a pixel coordinate processing script running on the Jetson Orin. This script evaluates whether the robotic arm is aligned with the detected fruit. If the arm is not centered, the base joint rotates either left or right, as directed by the Jetson Orin, to achieve alignment. Once the arm is centered, its base joint position is recorded, and the fruit is harvested. After the fruit is collected and stored in the SARDOG's bin, the robot advances to detect and harvest the next fruit.

Figure 3 presents the block diagram outlining the hardware architecture of the smart harvesting system. The system is powered by two onboard computers: a Raspberry Pi 5 and an NVIDIA Jetson Orin, which communicate through 5V relays connected to their GPIO headers. The Jetson Orin is responsible for powering the USB camera, which captures real-time video footage that is processed by the AI model for fruit detection. Once the fruit is detected, the Jetson Orin sends directional and verification signals, in the form of Boolean values, to the relay modules.

The directional command relay sends a HIGH signal to instruct the robotic arm to adjust its position to the right, and a LOW signal to move it to the left. The verification

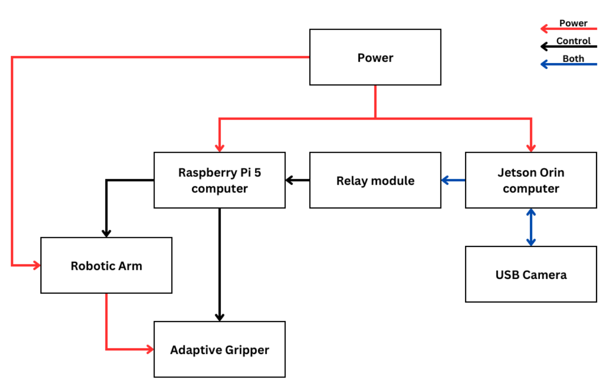

Fig. 3: Hardware block diagram.

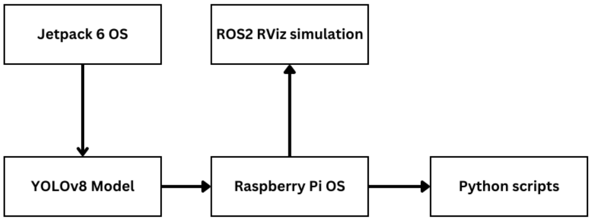

Fig. 4: Software block diagram.

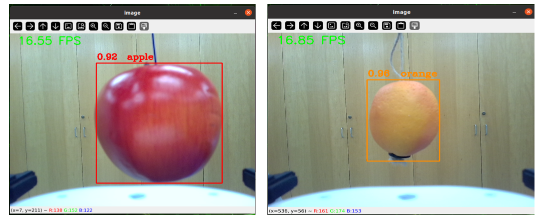

Fig. 5: Drawing bounding boxes around detected fruits.

command relay sends a HIGH signal to confirm that the arm is centered on the fruit and requires no further adjustment. If the arm is not centered, the relay remains LOW, instructing the system to continue reading input from the directional command relay.

Relays are used instead of direct GPIO connections due to voltage differences: the Raspberry Pi's GPIO header connects 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 aligned with the fruit, it transmits angular values to control the servos in the robotic arm and adaptive gripper.

Power for the system is supplied by SARDOG's power distribution board, connected to two 24V batteries. This board delivers 24V to the robotic arm and gripper and steps down the voltage to 5V for both onboard computers through 24V, 12V, and 5V converters.

B. Software System Overview

The Raspberry Pi 5, running the latest Debian Raspberry Pi Desktop Operating System, controls the servos in the robotic arm via an integrated hat board. We developed two primary Python scripts to handle the complex kinematics required for the harvesting process.

The first script, referred to as the analysis script, extends the arm to its maximum reach and then pans it horizontally. As the arm moves laterally, it reads input from relays controlled by the Jetson Orin. Depending on the Boolean value sent by the directional relay, the arm adjusts its position to the right or left to center itself on the detected fruit. Once centered, the second relay outputs a HIGH value, and the arm's calibrated start position is saved.

The second script governs the arm's picking motions, which are specifically tailored for various harvesting scenarios. These motions are defined by angular values sent to the six servos in the robotic arm, allowing precise and efficient fruit harvesting. The software block diagram is shown in Fig. 4.

The NVIDIA Jetson Orin, operating on the Ubuntu-based Jetpack 6 Operating System, manages all image processing tasks for the harvesting system. Leveraging the trained YOLOv8 model and the OpenCV Python library, the system detects, classifies, and localizes fruits in real-time using USB camera footage. The YOLOv8 model generates bounding boxes around the detected fruits based on pixel coordinates.

In our harvesting script, the center of each bounding box is calculated, as illustrated in Fig. 5. This center point is used to send directional and verification Boolean values to the relays. If the center point is to the left or right of the camera's frame, the Jetson Orin transmits the corresponding Boolean signal to the relays. When the center point aligns with the middle of the camera frame, a HIGH signal is sent to the verification relay, indicating the arm is correctly positioned to harvest the fruit.

During development, virtual simulation of the robotic arm's joint movements was critical for planning the kinematics and ensuring the mounted camera remained stable throughout the process. To achieve smooth and effective harvesting motions, the six servos had to operate at varying speeds to reach their final positions simultaneously. Using ROS2 Foxy, we simulated these joint motions. The RViz simulation environment developed by Elephant Robotics, shown in Fig. 6, allowed us to optimize the harvesting process.

The horizontal position calibration system, combined with this simulation, eliminated the need for coordinate control of the robotic arm. Coordinate control requires complex backend computations, as it converts a single coordinate into motions for all joints. In fruit harvesting, it is important for the arm's end effector to avoid branches and leaves as it reaches for the fruit, reducing the risk of damage. The relative angular control approach we implemented offers full control over every stage of the robotic arm's movements, allowing the end effector to be guided carefully around obstacles.

In contrast, systems relying on coordinate control automate joint movements toward a final point, often reducing customization and precision in handling [2].

III. Detection System Development

A. Dataset Compilation and Formatting

The master dataset was custom-compiled using unlabeled datasets from Kaggle. The dataset includes eight types of tree fruits, as shown in Fig. 7. To train the YOLOv8 model,

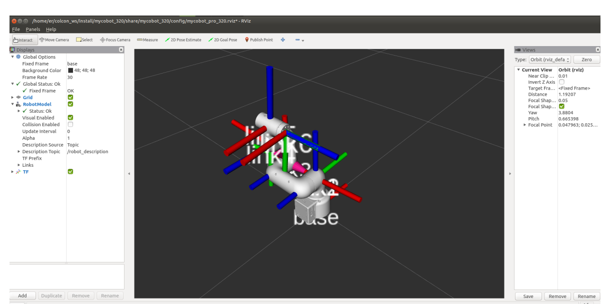
Fig. 6: ROS2 RViz joint control simulation environment.

the dataset was annotated using Roboflow to create bounding boxes for each fruit.

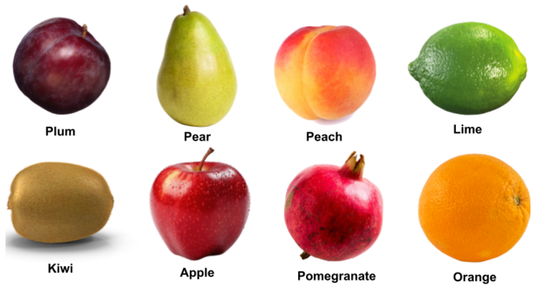

Fig. 7: Fruits in the master dataset.

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 such as ResNet50 and GoogleNet. This level of precision is crucial for the smart harvesting system, where the robotic arm must accurately center itself on the detected fruit, necessitating precise pixel coordinate data from the image frame. The training of our model was conducted using Google Colab, utilizing its accelerated GPU performance with an NVIDIA T4 Tensor Core. Initially, we ran the model for only 8 epochs, which resulted in inconsistent accuracy. However, after extending the training to 25 epochs, the model's performance significantly improved.

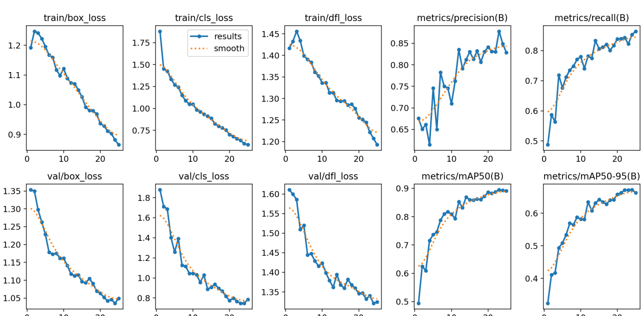

Fig. 8: Training and validation results.

The training results, depicted in Fig. 7, highlight the model's progress using our custom dataset. The overall training duration was 56 minutes, averaging about 2 minutes per epoch. By the 20th epoch, the model's predictions had stabilized at an 88% validation accuracy, indicating that it could reliably differentiate between eight different types of tree fruits. While the primary goal was to detect and localize fruit in general, these results confirm that the model achieves high accuracy in performing this task.

The confusion matrix shown in Fig. 9 visualizes the classification performance of the deep learning model. Notably, the model demonstrated higher accuracy in identifying oranges and apples compared to other fruits. This discrepancy is likely due to the greater number of images for these fruits in the master dataset, allowing the model more training data for these specific classes.

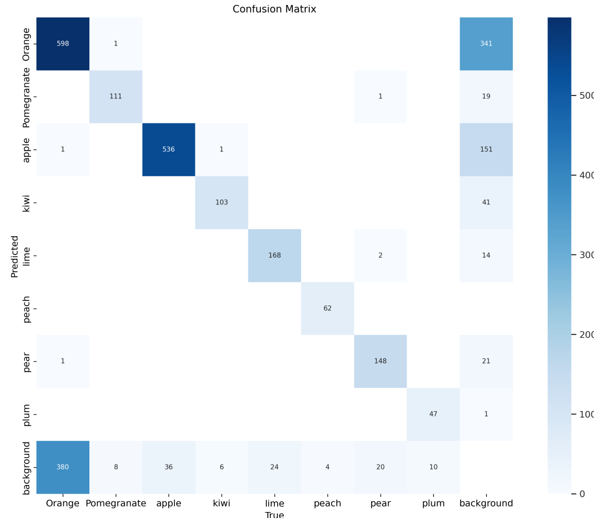

Fig. 9: Confusion matrix.

For more precise fruit-specific classification, further improvements—such as a larger, more balanced dataset and extended training times—would be required. However, for general fruit detection purposes, the current model performs adequately.

Fig. 10: Validation batch example.

Additionally, the validation results, illustrated in the collage of fruits, shown in Fig. 10, reflect the robustness of the model across various scenarios. The master dataset included a variety of image perspectives, ranging from individual fruits to clustered groups and fruits partially obscured by branches. When the fruit was clearly visible, the model consistently achieved a confidence rating of around 90%. In cases where the fruit was partially obscured, confidence ratings ranged between $60-80\%$, showcasing the model's capacity to handle complex real-world scenarios.

The results demonstrate the system's ability to accurately classify and localize fruits in real-time, making it suitable for precision agriculture applications.

IV. SARDOG Integration and Testing

A. Integration

To integrate the smart harvesting system into the SARDOG platform, several key modifications were made. The narrow base of the robotic arm presented challenges for securing it to a metal rod, necessitating the design of a custom wooden base. This wooden structure was fixed to the robotic arm using four long bolts, providing adequate space to accommodate additional hardware components. Once the platform was fully assembled, four U-bolts were used to attach the base securely to the leftmost rod of the SARDOG frame. This flexible setup allowed for easy attachment and detachment, facilitating the transport of the SARDOG through narrow passageways. The standalone smart harvesting system, prior to its integration with the SARDOG platform, is depicted in Fig. 11.

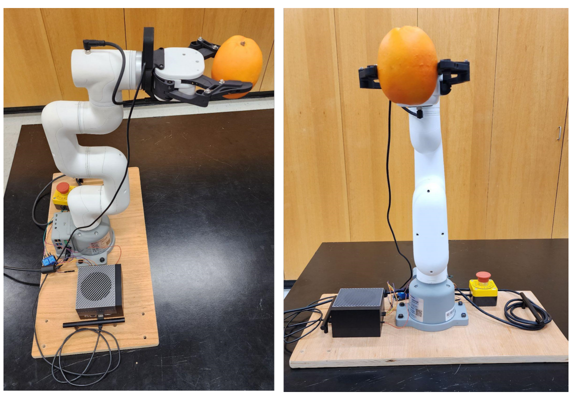

Fig. 11: Smart harvesting system by itself.

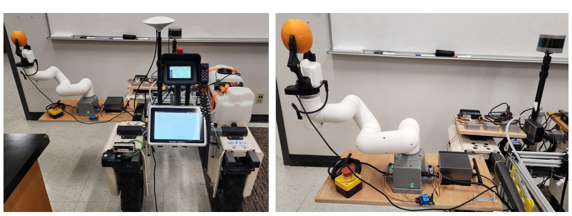

Fig. 12: Smart harvesting system integrated onto the SARDOG robot.

B. Testing and Experimentation

As illustrated in Fig. 12, positioning the robotic arm away from the side of the robot improved accessibility to hanging fruits. It became evident that widening the SARDOG frame to align with the width of orchard paths significantly enhanced system access. This adjustment was necessary because positioning the robot closer to the edge of the path was impractical, as other onboard systems required the robot to remain centered.

The system performed optimally when targeting fruits located on the outer edges of trees but faced some challenges in reaching fruits closer to the tree's interior. Extending the robotic arm could resolve this issue. Nevertheless, the YOLOv8 model successfully detected and classified fruits even in deeper sections of the trees. The arm's ability to execute both pulling and twisting motions enabled it to harvest oranges and plums without causing damage, as oranges require a twisting action, while plums benefit from a pulling motion.

V. Conclusion

This paper presents an autonomous smart harvesting system designed to optimize fruit-picking in precision agriculture. The integration of a six-degrees-of-freedom robotic arm with the SARDOG platform, powered by a YOLOv8 deep learning model, enables real-time detection and harvesting of various tree fruits. The system reduces labor demands while increasing operational efficiency, marking a significant step forward in agricultural automation. Future work will focus on improving the system's reach and adaptability to different orchard configurations and fruit types.

For a visual demonstration of our harvesting robot in operation, a video showcase is available via this link [12].

VI. Acknowledgment

The work was partially supported by the Economic Development Administration (EDA) Project Grant 077907908, Fresno-Merced Future of Food Innovation (F3) Coalition.

References

[1] H. Kulhandjian, B. Irineo, J. Sales, and M. Kulhandjian, "Low-cost tree health categorization and localization using drones and machine learning," in *2024 International Conference on Computing, Networking and Communications (ICNC)*, 2024, pp. 296–300.

[2] T. Yoshida, Y. Onishi, T. Kawahara *et al.*, "Automated harvesting by a dual-arm fruit harvesting robot," *Robomech Journal*, vol. 9, no. 1, p. 19, 2022.

[3] G.-J. Horng, M.-X. Liu, and C.-C. Chen, "The smart image recognition mechanism for crop harvesting system in intelligent agriculture," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2766–2781, 2020.

[4] R. K. Megalingam, N. Vignesh, V. Sivanantham, N. Elamon, M. S. Sharathkumar, and V. Rajith, "Low cost robotic arm design for pruning and fruit harvesting in developing nations," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5.

[5] Z. Yu, C. Lu, Y. Zhang, and L. Jing, "Gesture-controlled robotic arm for agricultural harvesting using a data glove with bending sensor and optitrack systems," *Micromachines*, vol. 15, no. 7, p. 918, 2024.

[6] N. R. N. Zadeh, H. Hashimoto, D. Raper, E. Tanuyan, and M. S. Bruca, "Autonomous smart farming system using flann-based feature matcher with robotic arm," in *AIP Conference Proceedings*, vol. 2502, no. 1. AIP Publishing, 2022.

[7] E. Vrochidou, V. N. Tsakalidou, I. Kalathas, T. Gkrimpizis, T. Pachidis, and V. G. Kaburlasos, "An overview of end effectors in agricultural robotic harvesting systems," *Agriculture*, vol. 12, no. 8, p. 1240, 2022.

[8] Roboflow, "Roboflow: Organize, label, and prepare your image data for computer vision," https://roboflow.com/, 2024, accessed: 2024-10-14.

[9] F. Salim, F. Saeed, S. Basurra, S. N. Qasem, and T. Al-Hadhrami, "Densenet-201 and xception pre-trained deep learning models for fruit recognition," *Electronics*, vol. 12, no. 14, p. 3132, 2023.

[10] M. O. I. Mureșan, "Fruits-360: A dataset of images containing fruits, vegetables, and nuts," https://www.kaggle.com/datasets/moltean/fruits, 2018, accessed: 2024-10-14.

[11] H. Kulhandjian, Y. Yang, and N. Amely, "Design and implementation of a smart agricultural robot bulldog (sardog)," in *2024 International Conference on Computing, Networking and Communications (ICNC)*, 2024, pp. 767–771.

[12] H. Kulhandjian and N. Amely, "AI-Powered Fruit Harvesting System Using a Robotic Arm for Precision Agriculture Video Demo," https://drive.google.com/file/d/1Y0u87O6fjBmaY6MrZDxUL0JBOgEP-89n/view?usp=sharing, 2024, accessed: 2024-10-19.