



# Article In-Depth Evaluation of Automated Fruit Harvesting in Unstructured Environment for Improved Robot Design

Sadaf Zeeshan <sup>1,\*</sup>, Tauseef Aized <sup>2</sup> and Fahid Riaz <sup>3,\*</sup>

- <sup>1</sup> Mechanical Engineering Department, University of Central Punjab, Lahore 54590, Pakistan
- <sup>2</sup> Mechanical Engineering Department, University of Engineering and Technology, Lahore 54590, Pakistan; tauseef.aized@uet.edu.pk
- <sup>3</sup> Mechanical Engineering Department, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates
- \* Correspondence: sadaf.zeeshan@ucp.edu.pk (S.Z.); fahid.riaz@adu.ac.ae (F.R.)

Abstract: Using modern machines like robots comes with its set of challenges when encountered with unstructured scenarios like occlusion, shadows, poor illumination, and other environmental factors. Hence, it is essential to consider these factors while designing harvesting robots. Fruit harvesting robots are modern automatic machines that have the ability to improve productivity and replace labor for repetitive and laborious harvesting tasks. Therefore, the aim of this paper is to design an improved orange-harvesting robot for a real-time unstructured environment of orchards, mainly focusing on improved efficiency in occlusion and varying illumination. The article distinguishes itself with not only an efficient structural design but also the use of an enhanced convolutional neural network, methodologically designed and fine-tuned on a dataset tailored for oranges integrated with position visual servoing control system. Enhanced motion planning uses an improved rapidly exploring random tree star algorithm that ensures the optimized path for every robot activity. Moreover, the proposed machine design is rigorously tested to validate the performance of the fruit harvesting robot. The unique aspect of this paper is the in-depth evaluation of robots to test five areas of performance that include not only the accurate detection of the fruit, time of fruit picking, and success rate of fruit picking, but also the damage rate of fruit picked as well as the consistency rate of the robot picking in varying illumination and occlusion. The results are then analyzed and compared with the performance of a previous design of fruit harvesting robot. The study ensures improved results in most aspects of the design for performance in an unstructured environment.

**Keywords:** fruit-harvesting robot; rapidly exploring random forest star; convolutional neural network; visual servoing

# 1. Introduction

Orchards, characterized by their diverse and irregular landscapes, pose distinctive challenges for modern machines, especially when it comes to deploying robotic technologies. Unlike controlled and structured environments, such as greenhouses, these natural settings are dynamic and often unpredictable [1]. When deploying advanced technology, such as robots, in these settings, they face obstacles like occlusion, shadows, poor illumination, and various environmental factors. Therefore, it is imperative to carefully account for these challenges during the design phase of harvesting robots.

Fruit-harvesting robots represent a significant advancement in agricultural technology with far-reaching implications for the world. These robots offer numerous benefits that address critical challenges facing the agricultural industry. First, they alleviate the labor shortage issue prevalent in many regions, particularly where manual fruit picking is physically demanding and labor-intensive. By automating the harvesting process, these robots ensure a reliable and efficient method of fruit collection, reducing dependency on human labor and mitigating associated costs. Additionally, fruit harvesting robots



Citation: Zeeshan, S.; Aized, T.; Riaz, F. In-Depth Evaluation of Automated Fruit Harvesting in Unstructured Environment for Improved Robot Design. *Machines* 2024, 12, 151. https://doi.org/10.3390/ machines12030151

Academic Editors: Salik Ram Khanal and Manoj Karkee

Received: 24 January 2024 Revised: 14 February 2024 Accepted: 20 February 2024 Published: 22 February 2024



**Copyright:** © 2024 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) license (https:// creativecommons.org/licenses/by/ 4.0/). enhance productivity and yield consistency by operating tirelessly and with precision, optimizing the harvesting process to maximize crop output. Moreover, they contribute to sustainability efforts by minimizing damage to crops and reducing food waste through gentle and selective harvesting techniques. By revolutionizing fruit harvesting practices, these robots pave the way for increased efficiency and profitability in the agricultural sector, ultimately benefiting both producers and consumers worldwide.

Fruit harvesting robots stand at the forefront of technological innovation, offering a revolutionary solution to amplify productivity and mitigate the dependency on manual labor for arduous and repetitive harvesting tasks. These automated robots are not only capable of streamlining the harvesting process but also possess sophisticated features that empower them to navigate and function seamlessly in intricate and ever-changing environments, particularly in orchards. Through thorough design considerations, these robots are equipped with advanced sensor systems, computer vision, and artificial intelligence, enabling them to adeptly tackle challenges such as occlusion, shadows, and insufficient lighting. Addressing these hurdles in the design phase allows these robots to optimize their performance and reliability, ensuring they can discern fruits amidst complex vegetation, navigate through obstructed spaces, and operate efficiently under various lighting conditions. The result is a cutting-edge technology that not only enhances agricultural efficiency but also marks a significant leap toward sustainable and autonomous agricultural practices.

This study uses a design of a lightweight fruit-harvesting robot using visual servoing integrated with a convolutional neural network (CNN) to improve the fruit harvesting of orange fruit. It further uses improved rapidly exploring random tree algorithm (RRT\*) to optimize the path planning of the robot. This paper separately analyzes and evaluates fruit picking performance with occlusion and without occlusion of fruit and leaves and with varying illumination. Hence, the novelty of this paper lies in the following aspects:

- 1. Developing an efficient orange harvesting robot featuring an optimized path-planning algorithm that utilizes an enhanced RRT\* model, ultimately improving the fruit pick-up time and testing it in a real-time scenario.
- Creating an improved fine-tuned fruit detection model employing CNN, seamlessly integrated with position visual servoing, to enhance the precision of robots identifying high-quality oranges.
- 3. Conducting a comprehensive performance evaluation of the harvesting machine across five domains, including precise fruit detection, efficient fruit-picking time, success rate of fruit picking, damage rate of picked fruit, and consistency of robot performance under varying illumination and occlusion conditions.

The paper is structured as follows:

Section 1 provides an overview of the study, covering its background, motivation, and contributions. In Section 2, a comprehensive literature analysis is conducted, focusing on the current state of the art in fruit-harvesting robots and the foundational principles supporting the study. Section 3 presents the design and methodology used in the development of the fruit-harvesting robot. This includes a detailed examination of the design structure, simulation model, algorithms, control system, and experimental setup. In Section 4, the experimental results are thoroughly scrutinized following extensive testing of the robot across various parameters such as detection accuracy, pick time, success rate, fruit damage rate, and consistency of performance. Furthermore, a comparative analysis is conducted between the performance of this robot and recent studies in the field of fruit-harvesting robotics. Finally, Section 5 concludes the paper by summarizing the key findings of the study and offering insights into potential future research directions.

### 2. Literature Analysis

In the ever-evolving landscape of agricultural robotics, the prospect of commercially viable fruit harvesting robots has remained a distant prospect, awaiting a breakthrough. Despite strides in automation, the complexities inherent in dynamic harvesting environments have proven to be a formidable challenge. However, the integration of state-of-the-art

convolutional neural network (CNN) models, integrated with visual servoing, presents a paradigm shift in redefining the capabilities of these robotic systems. This synergy, particularly when supported by the rigorous training of custom datasets, holds the key to substantially improving fruit detection accuracy. Additionally, the strategic deployment of optimized path-planning algorithms emerges as a cornerstone in not only refining precision but also streamlining the overall cycle time of the robotic harvesting process. This convergence of technological advancements signals a pivotal juncture, potentially propelling fruit-harvesting robots from the realms of experimentation to the threshold of commercial viability.

Many agricultural machines have been designed for improving agricultural tasks replacing manual operations [2]. The fruit-harvesting robot (FHR) is an important automated machine for improving the productivity of agricultural produce. However, for it to be commercially viable, it needs to be accurate, fast, and efficient. The pick-up time needs to be minimal and the success rate for picking quality fruits should be maximal. The pick-up rate depends on obstacle-free path planning and the shortest path adopted by the robot to grasp the fruit. The successful picking of quality oranges depends on the proper detection of the ripe fruit and the gripper being able to pick up fruit without damage. Hence, this study aims to design an efficient fruit-harvesting robot that addresses these factors.

For the path planning of the robot, many optimization algorithms have been used. Sadaf et al. [3] compared the most utilized path-planning algorithms for fruit-harvesting robots, namely probabilistic roadmap (PRM), A\*, rapidly exploring random tree (RRT), and improved rapidly exploring random tree (RRT\*) algorithms. After thorough simulation testing, it was found that RRT\* performed better by 21% than other common path-planning algorithms. This was also verified by Zhang et al. [4] who harvested tomatoes using RRT\* with path length reduced by 24%. Similarly, Wang et al. [5] concluded that for a fruit-harvesting robot, harvesting time was reduced by 40–60% by using RRT\*. Moreover, Lehnert et al. [6] picked sweet pepper using RRT\* successfully avoiding most obstacles. Furthermore, Wei et al. [7] proposed that improved RRT\* avoided obstacles and was favorable for path planning in unstructured environment.

Fruit detection is an important aspect of fruit harvesting robots. The correct detection of ripe fruit is only possible with an accurate target fruit detection algorithm. Convolutional neural network (CNN) has shown great advantage over other object detection methods due to its greater adaptability, ability to learn hierarchical representations of features, and its robust nature. Wan et al. [8] used faster R-CNN for the multiple detection of fruits by means of an existing Fruits 360 dataset. Also, Zhang et al. [9] were able to detect apples for picking using CNN with an accuracy of 99.4%. Gao et al. [10] was able to detect apples with an accuracy of 91.67%. The CNN algorithm was also used for the detection of tomatoes by Lui et al. [11]. Furthermore, Momeny et al. [12] used CNN to detect cherry fruit with a detection accuracy of 94%. CNN was also used for the detection of kiwi fruit with an accuracy of 89.6% [13]. Sadaf et al. [14] detected oranges using CNN with accuracy of 93.8%. Similarly, Yin et al. [15] detected grape clusters with an accuracy above 90%, whereas Liu et al. [16] detected citrus fruit with an accuracy of 95.35%.

Visual servoing deploys visual feedback to regulate the motion of the robot. It uses images from cameras to compute control signals for the robot's actuators. The integration of CNN with visual servoing influences the strengths in image processing and feature extraction to enhance the precision and adaptability of the visual control system. Park et al. [17] used visual servoing for cucumber harvesting in a greenhouse environment with a detection speed of 16–23 fruits per second. Similarly, Li et al. [18] integrated a hybrid visual servoing control method for fruit harvesting with an average harvesting time of 9.4 s and a success rate of 96.25%. Shi et al. [19] also deployed visual servoing to improve the repeated position accuracy of tomato fruit picking. A total of 93% of fruits were successfully picked. Moreover, Liu et al. [20] used visual servoing and were able to harvest grape clusters in 6.5 s with a success rate of 83.3%.

The main factors that affect fruit detection are occlusion, illumination, orientation, and the overlapping of fruit. Hence, to be able to correctly predict the correct success rate and pick-up time of fruit, testing should be conducted considering the overlapping of fruits, illumination, and orientation of fruit. Many studies in the recent past have conducted experimental testing in laboratories and the field to determine the success rate and pick-up duration of the fruit harvest by the robot. Xiong et al. [21] picked strawberries from a farm with a success rate up to 97.1% and a pick-up speed of 6.1 s. Similarly, Wang et al. [22] tested apple harvesting in an orchard with a success rate of 70.77%. Moreover, Yin et al. [23] harvested citrus with a successful picking rate of 87.2% and a cycle duration of 10.9 s. Williams et al. [24] also harvested kiwi with a success rate of 86% and cycle duration of 2.78 s. Also, Lili et al. [25] harvested tomatoes in a greenhouse with a detection rate of 99%, success rate of 86%, and cycle time of 15 s. Table 1 shows the recent fruit detection algorithms used in fruit-harvesting robots and their results.

Reference	Fruit	Year	<b>Classification Algorithm</b>	Results
Takeshi Yoshida [26]	tomatoes	2018	SVM	Success rate: 95%
Tao Li [27]	apples	2022	CNN	Errors reduced by 59%
Hamzeh Mirhaji [28]	oranges	2021	YOLO	Accuracy 90.8% Precision: 91.23% Recall: 92.8%
Hanwen Kang [29]	apples	2020	DasNet v-1	Accuracy: 90% Recall: 82.6% F1 score: 0.85
Zan Wang [30]	tomatoes	2022	CNN	mAP: 96.14%
P.K. Sekharamantry [31]	apples	2023	YOLO	Precision: 87% F1 score: 0.98
Lin Hunag [32]	oranges	2023	IGA-SVM	Accuracy: 98%
T. Hu [33]	apples	2023	YOLO	mAP: 94% F1 score: 0.93
Bayu Alif Farisqi [34]	guava	2022	CNN	mAP: 88% F1 score: 0.89
Jiaxing Xie [35]	litchi	2022	YOLO	Precision: 87.1%
T. Yuan [36]	tomatoes	2020	SSD	Precision: 98.85%
G. Coll-Ribes [37]	grapes	2023	CNN	mAP: 94.9%

Table 1. Fruit detection algorithm used in recent studies.

Table 2 displays the path optimization algorithms tested by fruit-harvesting robots. The main features of the selected algorithms are also highlighted for the relevant fruit.

Convolutional neural networks (CNNs) are a type of deep learning architecture specifically designed for processing grid-like data for tasks like image classification. They utilize mathematical operations such as convolutions and pooling to extract hierarchical features from input images, followed by fully connected layers for classification or regression. The convolution operation applies a filter (kernel) to an input image to produce a feature map. Mathematically, it is represented as:

$$Out(x,y) = \sum_{i-imin}^{imax} \sum_{j-jmin}^{jmax} f(i,j) g(x-i, y-i)$$
<sup>(1)</sup>

where *f* is the input image or feature map, g is the convolutional filter, and Out(x, y) is the output feature map at position (x, y).

Author	Year	Fruit	Algorithm	Salient Features
Z. Qin [38]	2021	grapes	RRT*	Rewiring, optimized path
G. Lin [39]	2021	guava	DDPG	Versatile, efficient
S. Kothiyal [40]	2021	orange	DDPG	Sample efficiency, off-policy algorithm
W. Wencheng [41]	2021	grapes	RRT	Easy implementation, non-optimal
A. Zahid [42]	2020	apple	RRT	Random samples, suitable for real-time
C. Lehnert [6]	2020	sweet pepper	RRT*	Efficient, optimal path
P. Kurtser [43]	2020	sweet pepper	RRT	Suitable for real-time, non-optimal
P. Kurtser [43]	2020	sweet pepper	GA	Optimal path, flexible, adaptive
Y. Liu [23]	2019	tomato	ACO	Optimal path, robust
H. Sarabu [44]	2019	apple	RRT	Efficient, suitable for real-time
L. Wang [45]	2019	apple	BSO	Global optimization, robust
Magalhães [46]	2019	grapes	BiT RRT	Efficient, complex implementation

Table 2. Path optimization algorithm used in fruit harvesting.

After the convolution operation, an activation function is typically applied element wise to introduce nonlinearity. Common activation functions include ReLU (rectified linear unit), sigmoid, and tanh. Pooling operations like max pooling or average pooling reduce the spatial dimensions of the feature maps. Max pooling, for example, takes the maximum value within localized regions as shown in Figure 1 below.



Figure 1. Max pooling operation.

In the final layers of a CNN, fully connected layers perform weighted sums of the inputs followed by activation functions. Mathematically, this can be represented as:

$$Output = \sigma(Wx + b) \tag{2}$$

where  $\sigma$  is the activation function, W is the weight matrix, x is the input vector, and b is the bias vector. These equations form the backbone of CNNs and are used iteratively across different layers to learn hierarchical representations of the input data.

Determining hyperparameters for a convolutional neural network (CNN) involves a combination of empirical experimentation and domain knowledge. Initially, common defaults are used, including typical values for learning rate, batch size, number of epochs, activation functions, and optimizers. Experimentation with the learning rate is essential to find a balance between convergence speed and stability. Batch size is adjusted to ensure stable training without excessive memory usage. Network architecture, including layer depth, filter sizes, and neuron counts, is explored, with popular architectures serving as starting points. Regularization techniques using dropout help prevent overfitting. Validation or cross-validation is employed to assess model performance across different hyperparameter configurations. Automated hyperparameter tuning methods using Bayesian optimization can systematically explore the hyperparameter space for optimal configurations. By iteratively adjusting these parameters and evaluating their impact, the CNN is fine tuned to achieve optimal performance for the given task and dataset. Visual servoing is a technique used in robotics where control commands are computed based on visual feedback to guide the robot toward a desired state or trajectory. It often involves estimating the pose of an object or scene using vision sensors and then computing control commands to minimize the error between the desired and current poses. Mathematically, visual servoing can be represented by the following equation:

1

$$\iota = \lambda \mathbf{J}' + \mathbf{e} \tag{3}$$

where *u* represents the control input in terms of joint velocities,  $\lambda$  is a gain parameter, J' is the pseudo-inverse of the Jacobian matrix J, and e denotes the error between desired and current poses. The pseudo-inverse allows for the control of redundant degrees of freedom in the system. This control law enables the robot to adjust its velocities based on visual feedback, aiming to minimize the error between the desired and current poses. Adjusting the gain parameter  $\lambda$  allows for tuning the response of the control system to achieve desired performance characteristics.

Rapidly exploring random trees (RRT\*) is a popular motion-planning algorithm used in robotics to efficiently explore high-dimensional configuration spaces. It incrementally builds a tree rooted at the initial configuration by randomly sampling the configuration space and connecting the nearest tree node to the sampled point. Euclidean distance is used for estimating the distance between the existing current node and the target node to reach fruit.

$$l(n) = \sqrt{\left(x_n - x_{fruit}\right)^2 + \left(y_n - y_{fruit}\right)^2} \tag{4}$$

where  $(x_n, y_n)$  are the center coordinates of the point at the current position and (x fruit, y fruit) are the center of the target fruit position.

## 3. Design and Methodology

## 3.1. Methodology of Development of Fruit Harvesting Robot

The flowchart in Figure 2 below shows the development and working process of the orange-harvesting robot for the given research. The process of fruit harvesting begins with the RGB-D camera capturing high-resolution images of the orchard, which are then fed into a pre-trained CNN algorithm. The CNN algorithm detects the oranges and makes a bounding box around them. The fine-tuned algorithm ensures the detection of ripe fruit only. The localization of the fruit is performed using the RGB-D camera sensor, which tells the x-, y-, and z-axis location of the nearest fruit. The inverse kinematics computes a path from end-effector to the fruit, while the path-planning algorithm, utilizing the RRT\* algorithm, optimizes the trajectory, minimizing the path length. The continuous assessment of the robot's position and orientation against the desired pose facilitates error calculation, refining the accuracy of the harvesting process. Visual servoing uses feedback control loops to adjust the robot's motion based on visual information obtained from its environment. By continuously analyzing images captured by onboard cameras, the system can make real-time adjustments to the robot's position and orientation. This iterative process ensures that all ripe fruits are efficiently picked within a single harvesting cycle, optimizing the yield and enhancing overall productivity.



Figure 2. Flowchart of the orange-harvesting robot's improved design.

# 3.2. Design of Fruit Harvesting Robot

The five-degree-of-freedom orange-harvesting robot is made with an aluminum frame with a maximum length of 959.61 mm with weight of only 15.5 kg. The end effector comprises a three-finger silicon gripper. The RGB-D Kinect camera is mounted separately in a fixed position. The CAD model is designed on SolidWorks. The torques of each joint are computed, and actuator selection is conducted. Static analysis is followed by simulation to ensure the proper movement of the robot. Figure 3 below shows the CAD model and the actual model of the fruit-harvesting robot.



Figure 3. SolidWorks CAD model and actual model of a fruit harvesting robot.

Figure 4 below shows the simulation of the five degrees of freedom on Simulink on MATLAB. The simulation of the CAD model assists in analyzing the movement of the fruit harvesting robot. Hence, it aids in the path planning of the robot. The Simulink diagram for a five-degree-of-freedom (DOF) robot serves as a comprehensive virtual environment for modeling, simulating, and analyzing the robot's behavior and performance. It allows for the accurate representation of the robot's kinematics, enabling the testing of various control algorithms for motion regulation and trajectory planning. Through simulations, the diagram facilitates the validation of trajectories and supports the integration of the robot model with other systems and components.



Figure 4. Detailed model of the Simulink model of five DOF fruit-harvesting robotic arm.

### 3.3. Path Planning of Robot Using RRT\*

The proper success of the applied path-planning model can be measured by its ability to spot a path, evade obstructions and collisions, to form a path that is optimized, and have the shortest time for path recognition, as well as minimal computational time. RRT\* is an improved option of the RRT model that is precisely considered to explore the nearest route between two given selections in the configuration space. It is best known for discovering the shortest route, hence shortening the route travelled by the robot. It is a highly efficient algorithm as it discovers the vacant zones in a way that prevents the revisiting of the discovered areas. It can deal with dynamic settings well and evade obstructions when finding a route. Figure 5 below shows a sample demonstration of the path planning of RRT\* for fruit picking. The green line depicts the simulated path from the end effector to the target fruit, while blue arrow shows the planned path. The total computation time and pick-up time result in the total cycle time to pick the fruit.



Figure 5. Path determination using RRT\* algorithm.

The algorithm begins with the creation of a single node representing the initial configuration as seen in Figure 6. Subsequently, it iteratively expands the tree by randomly selecting points in the state space and extending the tree toward these points. Nodes are added to the tree based on the closest existing node, and the new node is connected to its nearest neighbors. The red dots show the iterative expansion of the tree by growing nodes in space. However, due to the rewiring features, only the shortest path is computed, that is shown in green. The steps for RRT\* are shown in the Algorithm 1 below.

### Algorithm 1 RRT\* R = (V, E)

Set up a <sub>init</sub> , a <sub>goal</sub>			
R 4	— Initialize Tree ();		
for	i = 1n do		
	$a_{rand}$ $\blacktriangleleft$ Sample (i);		
	$a_{near}$ $\leftarrow$ Nearest(R, $a_{rand}$ );		
	$a_{new} \leftarrow Steer(a_{near}, a_{rand});$		
	If ObstacleFree (a <sub>new</sub> ) then		
	$a_{close}$ $\blacktriangleleft$ Near (R, $a_{new}$ );		
	$a_{\min}$ $\leftarrow$ Choose parent ( $a_{close}, a_{near}, a_{new}$ );		
	R.addNodeEdge (a <sub>min</sub> , a <sub>new</sub> );		
	R.rewire();		
return			



Figure 6. The node formation process of RRT\* algorithm.

Path planning using the improved RRT\* algorithm is performed through the given steps. First, we create the environment map. We define the orchard environment, including the location of the fruit trees, obstacles, and any other relevant information that the robot needs to navigate. Then, we use this information to create a binary map. Next, we define the beginning and goal states. This is performed by marking the starting position and the goal location of the robot. This is where the robot starts and ends its path planning. Then, we initialize the RRT\* algorithm. This creates an initial random tree, with the start state as the root of the tree. Now, random samples are generated in the orchard environment and checked to ensure they are collision-free. Thereafter, the random samples are connected to the nearest point in the tree, creating new branches. Next, it is checked if the random sample is close to the goal state and that the path is collision-free and valid. This is followed by the optimization of the path. If a valid path is found, the path is optimized by removing unnecessary branches and smoothing the path to reduce the amount of turns and enhance its efficiency. This feature is called rewiring. The rewiring attribute permits the route to readapt itself on the basis of the shortest route. Rewiring is a method that comprises searching for fresh edges on the map to produce new routes, along with discarding routes that are no longer required. It allows the route to adjust to the variations in the setting. The procedure of rewiring assists regular improvement in the route, favoring the nearest route to be formed. Therefore, this technique aids in finding the most effective route by endlessly upgrading the track by discovering fresh, close-by nodes. As the number of iterations increases, the probability that a route shall be found also rises exponentially. Then, the process of generating random samples and connecting them to the tree until a valid path to the goal is found is repeated. Finally, we implement the optimized path on the robot, allowing it to navigate through the orchard and reach the goal. Figure 7 below summarizes the steps for the path planning process.



Figure 7. Path-planning process for RRT\*.

## 3.4. Fruit Detection

Fruit detection is performed by the fruit-harvesting robot through the convolutional neural network algorithm. The CNN is a model that acquires images and assigns them weights. Weights are the importance factors or biases that are applied to differentiate one object from another. It works by organizing their nodes into series of layers. The outcome from each layer is fed to the succeeding layer for further output generation after the processing of data. Training data and test data are taken as a ratio of 80:20. The data set is a custom-designed collection of oranges of different orientation, sizes, illumination, and fruit occlusion specifically from the fruit picking area of target orchards to increase its detection accuracy. As the data set is trained for a neural network, it uses a loss function that determines the inaccuracy margin of the actual output from the desired output. This information is then deployed to update the weights of the model.

Various convolutional filters are deployed to excerpt various features of the image. This is the filtering process that helps to improve the predictions of the results. The initial layer learns the basic detection filters like the edges of the fruit, shape of the fruit, etc. The middle layers learn further features for the detection of the object like the texture of the fruit. The last layers learn to detect various orientations, the full object as a whole, and the higher representative features of the object.

Strides inform how many pixels in the convolutional filter move every time it processes a group of pixels of the image. Very long strides can result in the slipping of important image features. Oftentimes, images are padded with empty pixels so that the generated feature map holds the same dimensions as the original image. The ReLU activation function is implemented to improve the image features and cut down the noise. Maximum pooling chooses the maximum values of a pixel of each scanning at every instance within the filter. The resulting image contains a maximum pooled value representation of the original image. Table 3 below shows the hyperparameters for the model.

Batch Size	64
Neuron number per layer	128
Number of hidden layers	4
Method of Optimization	RMSprop
Model loss function	Cross-entropy
Activation function	ReLU
Kernel size of filters	$3 \times 3$
Number of filters	64
Epoch Number	100

Table 3. Hyperparameters of the fine-tuned CNN model.

The positive and negative predicted statistics tell us the values of true positives and negatives along with false negatives and positives of our experimental data. The metrics are calculated to assess the results of the given algorithm. Figure 8 below shows the fruit detection process using Kinect RGB-D camera integrated with the CNN model. The indicators used to assess the performance of the algorithm are accuracy, precision, recall, and F1 score.



Figure 8. Fruit detection using CNN model using Kinect RGB-D camera.

### 3.5. Visual Servoing

Visual servoing is an effective technique used in robotics to control the motion of a robot based on visual feedback from a camera. For our fruit harvesting, a visual servoing system was deployed to guide the robot's end effector to a specific location to pick a fruit. The control system was integrated with the visual servoing system to ensure that the robot's motion was accurate and smooth. This involved using sensors, such as RGB-D cameras and encoders, to provide feedback on the robot's position and orientation, and then using this feedback to adjust the robot's motion in real time. The control system was also responsible for coordinating the motion of robotic arm joints in the robot to achieve the desired motion.

The control process of position-based visual servoing is used to obtain information from the camera sensor and compute control signals for output as shown in Figure 9 below. Once the position and orientation of fruit is determined, the control system uses this information to generate control signals that direct the robot toward the fruit and determine the best way to reach it by avoiding all obstacles. The encoder outputs a series of digital signals that determine the position and speed of the robotic arm. It provides real-time information that is used to accurately position the robotic arm at the required location. The eye-to-hand camera analyzes the image and determines the distance of the fruit from the gripper and camera using the triangulation method. Once the position of the object is determined and distance of fruit to gripper is computed, signals are generated, and the gripper moves toward the fruit and grasps it. The actuator shall need to be calibrated to ensure that it responds properly to the control signals. The control system of the fruit-harvesting robot utilizes visual servoing, leveraging a Raspberry Pi 4 and integrated drivers to orchestrate the motion of five actuators for precise harvesting operations. The Raspberry Pi 4 serves as the central processing unit, receiving real-time visual input from onboard cameras to guide the robot's actions. Through an image detection algorithm, the system analyzes the position, ripeness, and accessibility of fruits within the orchard. Using this information, the Raspberry Pi computes the optimal trajectory for the robot's five actuators, each strategically placed to execute harvesting maneuvers with accuracy and efficiency. Integrated drivers enable seamless communication between the Raspberry Pi and actuators, translating control commands into precise motor movements. This visual servoing approach ensures that the fruit-harvesting robot can autonomously navigate orchards, identify ripe fruits, and execute harvesting tasks with precision and also influence the computational power of the Raspberry Pi 4 for efficient control and decision making.

Required

Output +



Figure 9. Position of visual servoing control using eye-to-hand configuration.

# 3.6. Experimental Setup and Testing in Laboratory before Real-Time Evaluation

The fruit-harvesting robot was tested with 106 fruit samples. Harvesting situations were tested for occluded and non-occluded fruits. The test area for robot reachability was  $7 \times 7$  m without changing the robot base position. The test was conducted with an illumination of 400 lumens/m<sup>2</sup>. The sensor for fruit recognition was the Kinect RGB-D camera. Figure 10 shows the evaluation of the fruit-harvesting robot in the test environment.



Figure 10. Testing of fruit-harvesting robot in a laboratory.

# 4. Results and Discussion

4.1. Fruit Detection in Real-Time Testing

Oranges are detected in the orchards of Sargodha, Pakistan. Real-time testing is performed in the orchard of Kot Momin in Sargodha. Fruit harvesting took place in an orchard which contained 170 orange trees in 15 rows with an inter-row spacing of 5 m. The trees were 5–8 m high. The first evaluation is performed for orange detection, as shown in Figure 11. Figure 12 shows the field testing of the orange-harvesting robot. The accuracy,



Figure 11. Fruit detection testing of oranges in orchard.



Figure 12. Fruit picking process in orange orchard.



Figure 13. Evaluation results for fruit detection for the deployed CNN model.



Figure 14. Graph for model loss vs. epoch number for the CNN model for train and test data.

Evaluation parameters for fruit detection are based on accuracy, precision, recall, and F1 score. Using the numbers of true data (true positives and true negatives) and false data (false positives and false negatives), the accuracy, precision, recall, and F1 score are computed using formulas below.

$$Accuracy = (True_{pos} + True_{neg}) / (True_{pos} + True_{neg} + False_{pos} + False_{neg})$$
(5)

$$Precision = \frac{True_{pos}}{True_{pos}} + False_{pos}$$
(6)

$$Recall = \frac{True_{pos}}{True_{pos}} + False_{neg}$$
(7)

$$F1_{score} = \frac{2 \times precision \times recall}{precision + recall}$$
(8)

# 4.2. Cycle Time for Fruit Picking in Real-Time Testing

The cycle time of occluded fruit verses non-occluded fruit is shown in Figure 15 below. The average cycle time for non-occluded fruits was 7.9 s. The average cycle time for occluded fruits was 6.8 s. The overall average fruit-picking cycle time was 7.2 s. The cycle time is calculated by the following:

$$Cycle Time = \frac{(Total time taken for fruit picking)}{(No. of fruits harvested)}$$
(9)



Figure 15. Comparison of cycle time for orange picking with and without occlusion.

(10)

### 4.3. Success Rate in Real-Time Testing

The success rate of a fruit-harvesting robot hinges on the detection of the fruit, reaching for the fruit, and grasping of the fruit successfully. Figure 16 below shows the percentage of success of detection of fruit, reachability toward the fruit, grasping of the fruit, picking of the fruit, and percentage damage rate for the testing conducted for the harvesting of sample oranges. The figure below shows the comparison of harvesting success rates for occluded fruit as well as for no occlusion. Furthermore, Figure 17 below shows the comparison of harvesting success rates for all cases were tabulated and the average success rate of the fruit-harvesting robot for all cases came out to be 90.5%, calculated using the formula below.



Figure 16. Performance indicators for success rates of occluded and non-occluded fruit harvesting.





### 4.4. Damage Rate in Real-Time Testing

The often overlooked yet crucial parameter in evaluating the performance of a fruitharvesting robot is the damage rate, a factor that directly influences the quality of harvested fruit. Beyond merely picking the fruit, the robot must ensure the integrity of the fruit's skin and overall quality. Assessing the damage rate is vital, and the conducted tests revealed noteworthy insights. Specifically, occluded fruit picking exhibited a higher damage rate of 11%, while fruit harvested without occlusion displayed a lower rate of 6%, as shown in Figure 18 below. Similarly, the damage rate for varying illumination shows damage rate increases slightly with poor illumination as seen in Figure 19. The observed damages were predominantly attributed to inaccuracies in the gripping orientation of the end effector, leading to bruising or the inadvertent peeling of the fruit's top skin due to pulling at incorrect angles.



Figure 18. Comparison damage rate of fruit picking with and without occlusion.



Figure 19. Comparison of damage rate of fruit picking with varying illumination.

## 4.5. Consistency of Fruit Pick-Up in Real-Time Testing

To achieve commercial success, a fruit-harvesting robot must not only successfully pick fruit but also demonstrate consistent performance. Consistency is a key indicator of a well-designed and reliable model, ensuring the robot's ability to function reliably in diverse agricultural environments. This reliability is crucial for meeting the demands of precision agriculture, where a consistent and efficient fruit-harvesting process is essential for long-term success. The integration of reliability with efficiency becomes a hallmark for the sustained effectiveness of such robotic solutions in the agricultural sector. Figure 20 below shows a comparison of the consistency of the performance of the model for its fruit pick-up time without presence of occlusion and with presence of occlusion. It can be seen that the model overall performs consistently for both cases; however, amore consistent performance can be seen for fruit picking without occlusion.



Figure 20. Comparison of fruit picking consistency with and without occlusion.

### 4.6. Discussion

In this study, the integration of an improved convolutional neural network (CNN) model with position visual servoing has yielded promising results for the fruit-harvesting robot. The enhanced CNN model, trained on a curated custom dataset, demonstrated a substantial improvement in fruit detection accuracy within dynamic harvesting environments. This breakthrough not only addressed the persistent challenges associated with occluded conditions but also showcased the adaptability of the model, improving the overall accuracy of the model to 94% and precision to 98.2%.

Furthermore, the synergistic utilization of an optimized path-planning algorithm significantly contributed to the overall efficiency of the robotic harvesting process. By strategically navigating the robot through its operational space, the algorithm markedly reduced the cycle time of the entire harvesting operation with an average cycle time of 7.2 s. An average human picking time ranges from 8–12 s. This streamlined approach of the robot not only bolstered the precision of fruit picking but also rendered the system more time efficient, a crucial factor in the commercial viability of such robotic applications.

The assessment of the damage rate in a fruit-harvesting robot is a crucial aspect often underestimated in evaluating its overall performance. Beyond the primary task of fruit retrieval, preserving the quality and integrity of the fruit's skin becomes imperative. The significance lies in preventing bruising and ensuring that the fruit remains undamaged. The conducted tests highlighted the impact of occlusion, with a higher damage rate of 11% compared to a lower rate of 6% for fruit picked without occlusion. Notably, damages were attributed to inaccuracies in the gripping orientation of the end effector, causing bruising and slight peeling of the top skin due to incorrect pulling angles. This underscores the importance of assessing the damage rate as a key metric, emphasizing the need for precise gripping mechanisms to enhance the overall effectiveness of the fruit harvesting robot.

Similarly, this paper also emphasizes that success is not solely dependent on the robot's ability to successfully pick fruit; rather, a paramount factor is also its consistent performance. Consistency is particularly vital in the realm of precision agriculture, where the demand for a reliable and efficient fruit-harvesting process is paramount for long-term success. The model designed for a fruit-harvesting robot shows consistent results, particularly for non-occluded fruits.

Comparing the results with other recent field-tested fruit-harvesting robots, we find that Yin et al. [23] tested a citrus-harvesting robot in the actual field with a success rate of 87.2% and a pick up time of 10.9 s. Furthermore, Zhang et al. [47] harvested apples in the field using a four-degree-of-freedom robot using the CNN algorithm for fruit detection

having a success rate of 82.4% and a pick time of 6 s. The fruit detection accuracy was 90%. Moreover, Hu et al. [48] also tested a citrus-harvesting robot with a success rate of 90% and a pick up time of 15 s.

As we consider the implications of our findings, it becomes evident that this integration has the potential to significantly improve the fruit-harvesting process. The success observed in this study paves the way for future research and advancements, ultimately propelling the field toward widespread adoption and practical implementation. The scope of our study is confined to testing conducted in well-illuminated environments. However, for future investigations, we recommend expanding the study to encompass scenarios with varying levels of illumination.

### 5. Conclusions

Fruit-harvesting robots are an agricultural tool designed to increase the efficiency of production by accurately and quickly picking quality fruits. A fruit-harvesting robot contributes to sustainability by reducing reliance on manual labor, optimizing resource usage, and minimizing food waste through precise harvesting techniques, ultimately promoting efficiency and environmental responsibility in the agricultural sector. With the use of a lightweight fruit-harvesting robot having a custom-made dataset and improved CNN features integrated with position visual servoing, the accuracy of fruit detection is increased to 94%, with a precision rate of 98.2%, a recall rate of 95%, and an F1 score of 96.6%. The overall average cycle time of fruit picking is 7.2 s, with even less time taken for picking fruits without occlusion. Furthermore, the average success rate of fruit harvesting is 90.5%. The average damage rate is 8%, with damage mostly caused due to the wrong orientation of fruit picking, causing bruising, or due to the occlusion of fruits. These promising results underscore the viability and efficacy of integrating advanced robotic systems into mainstream agricultural practices. For future study, damage rate can be reduced, and grasping quality can be enhanced using improved sensors with the gripper and on the robot frame. Improved actuators can also help improve the speed of the picking process further.

Author Contributions: Conceptualization, S.Z. and T.A.; methodology, S.Z.; software, S.Z.; validation, S.Z., T.A. and F.R.; investigation, S.Z.; resources, T.A. and F.R.; writing original draft, S.Z.; writing—review and editing, S.Z., T.A. and F.R.; visualization, S.Z.; funding acquisition, T.A. and F.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors gratefully acknowledge the grant under letter PHEC/PICRA/20168/5 by the Punjab Higher Education Commission, Punjab, Pakistan, for this research.

**Data Availability Statement:** The dataset can be requested on a reasonable request by the mutual consent of the authors and the funding agencies.

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

### References

- Kootstra, G.; Wang, X.; Blok, P.M.; Hemming, J.; Henten, E.v. Selective Harvesting Robotics: Current Research, Trends, and Future Directions. *Curr. Robot. Rep.* 2021, 2, 95–104. [CrossRef]
- Wu, X.; Bai, J.; Hao, F.; Cheng, G.; Tang, Y.; Li, X. Field Complete Coverage Path Planning Based on Improved Genetic Algorithm for Transplanting Robot. *Machines* 2023, 11, 659. [CrossRef]
- 3. Zeeshan, S.; Aized, T. Performance Analysis of Path Planning Algorithms for Fruit Harvesting Robot. *J. Biosyst. Eng.* 2023, 48, 178–197. [CrossRef]
- 4. Zhang, Q.; Liu, F.; Li, B. A heuristic tomato-bunch harvest manipulator path planning method based on a 3D-CNN-based position posture map and rapidly-exploring random tree. *Comput. Electron. Agric.* **2023**, *213*, 108183. [CrossRef]
- Wang, Y.; Liu, D.; Zhao, H.; Li, Y.; Song, W.; Liu, M.; Tian, L.; Yan, X. Rapid citrus harvesting motion planning with pre-harvesting point and quad-tree. *Comput. Electron. Agric.* 2022, 202, 107348. [CrossRef]
- 6. Lehnert, C.; McCool, C.; Sa, I.; Perez, T. Performance improvements of a sweet pepper harvesting robot in protected cropping environments. *J. Field Robot.* **2020**, *37*, 1197–1223. [CrossRef]
- Wei, K.; Ren, B. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. Sensors 2018, 18, 571. [CrossRef]

- Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* 2020, 168, 107036. [CrossRef]
- Zhang, H.; Tang, C.; Sun, X.; Fu, L. A Refined Apple Binocular Positioning Method with Segmentation-Based Deep Learning for Robotic Picking. Agronomy 2023, 13, 1469. [CrossRef]
- Gao, M.; Ma, S.; Zhang, Y.; Xue, Y. Detection and counting of overlapped apples based on convolutional neural networks. J. Intell. Fuzzy Syst. 2023, 44, 2019–2029. [CrossRef]
- 11. Liu, S.; Zhai, B.; Zhang, J.; Yang, L.; Wang, J.; Huang, K.; Liu, M. Tomato detection based on convolutional neural network for robotic application. *J. Food Process Eng.* 2022, 46, e14239. [CrossRef]
- 12. Momeny, M.; Jahanbakhshi, A.; Jafarnezhad, K.; Zhang, Y.-D. Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. *Postharvest Biol. Technol.* 2020, *166*, 111204. [CrossRef]
- Williams, H.A.; 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]
- 14. Zeeshan, S.; Aized, T.; Riaz, F. The Design and Evaluation of an Orange-Fruit Detection Model in a Dynamic Environment Using a Convolutional Neural Network. *Sustainability* **2023**, *15*, 4329. [CrossRef]
- 15. Yin, W.; Wen, H.; Ning, Z.; Ye, J.; Dong, Z.; Luo, L. Fruit Detection and Pose Estimation for Grape Cluster–Harvesting Robot Using Binocular Imagery Based on Deep Neural Networks. *Front. Robot. AI* **2021**, *8*, 626989. [CrossRef] [PubMed]
- Liu, Y.-P.; Yang, C.-H.; Ling, H.; Mabu, S.; Kuremoto, T. A Visual System of Citrus Picking Robot Using Convolutional Neural Networks. In Proceedings of the 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018.
- 17. Park, Y.; Seol, J.; Pak, J.; Jo, Y.; Kim, C.; Son, H.I. Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector. *Comput. Electron. Agric.* **2023**, *212*, 108116. [CrossRef]
- 18. Li, Y.-R.; Lien, W.-Y.; Huang, Z.-H.; Chen, C.-T. Hybrid Visual Servo Control of a Robotic Manipulator for Cherry Tomato Harvesting. *Actuators* 2023, *12*, 253. [CrossRef]
- 19. Shi, Y.; Jin, S.; Zhao, Y.; Huo, Y.; Liu, L.; Cui, Y. Lightweight force-sensing tomato picking robotic arm with a "global-local" visual servo. *Comput. Electron. Agric.* **2023**, 204, 107549. [CrossRef]
- 20. Liu, J.; Liang, J.; Zhao, S.; Jiang, Y.; Wang, J.; Jin, Y. Design of a Virtual Multi-Interaction Operation System for Hand–Eye Coordination of Grape Harvesting Robots. *Agronomy* **2023**, *13*, 829. [CrossRef]
- 21. Xiong, Y.; Ge, Y.; Grimstad, L.; From, P.J. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *J. Field Robot.* **2019**, *37*, 202–224. [CrossRef]
- 22. Wang, X.; Kang, H.; Zhou, H.; Au, W.; Wang, M.Y.; Chen, C. Development and evaluation of a robust soft robotic gripper for apple harvesting. *Comput. Electron. Agric.* 2023, 204, 107552. [CrossRef]
- 23. Yin, H.; Sun, Q.; Ren, X.; Guo, J.; Yang, Y.; Wei, Y.; Huang, B.; Chai, X.; Zhong, M. Development, integration, and field evaluation of an autonomous citrus-harvesting robot. *J. Field Robot.* **2023**, *40*, 1363–1387. [CrossRef]
- 24. Williams, H.; Ting, C.; Nejati, M.; Jones, M.H.; Penhall, N.; Lim, J.; Seabright, M.; Bell, J.; Ahn, H.S.; Scarfe, A.; et al. Improvements to and large-scale evaluation of a robotic kiwifruit harvester. *J. Field Robot.* **2019**, *37*, 187–201. [CrossRef]
- Lili, W.; Bo, Z.; Jinwei, F.; Xiaoan, H.; Shu, W.; Yashuo, L.; Zhou, Q.; Chongfeng, W. Development of a tomato harvesting robot used in greenhouse. *Int. J. Agric. Biol. Eng.* 2017, 10, 140–149. [CrossRef]
- 26. Yoshida, T.; Fukao, T.; Hasegawa, T. Fast Detection of Tomato Peduncle Using Point Cloud with a Harvesting Robot. *J. Robot. Mechatron.* **2018**, *30*, 180–186. [CrossRef]
- 27. Li, T.; Feng, Q.; Qiu, Q.; Xie, F.; Zhao, C. Occluded Apple Fruit Detection and Localization with a Frustum-Based Point-Cloud-Processing Approach for Robotic Harvesting. *Remote Sens.* **2022**, *14*, 482. [CrossRef]
- Mirhaji, H.; Soleymani, M.; Asakereh, A.; Mehdizadeh, S.A. Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. *Comput. Electron. Agric.* 2021, 191, 106533. [CrossRef]
- 29. Kang, H.; Zhou, H.; Wang, X.; Chen, C. Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting. *Sensors* **2020**, *20*, *5670*. [CrossRef] [PubMed]
- 30. Wang, Z.; Ling, Y.; Wang, X.; Meng, D.; Nie, L.; An, G.; Wang, X. An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios. *Ecol. Inform.* **2022**, *72*, 101886. [CrossRef]
- 31. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516. [CrossRef]
- Huang, L.; Chen, Y.; Wang, J.; Cheng, Z.; Tao, L.; Zhou, H.; Xu, J.; Yao, M.; Liub, M.; Chen, T. Online identification and classification of Gannan navel oranges with Cu contamination by LIBS with IGA-optimized SVM. *Anal. Methods* 2023, 15, 738–745. [CrossRef] [PubMed]
- 33. Hu, T.; Wang, W.; Gu, J.; Xia, Z.; Zhang, J.; Wan, B. Research on Apple Object Detection and Localization Method Based on Improved YOLOX and RGB-D Images. *Agronomy* **2023**, *13*, 1816. [CrossRef]
- 34. Farisqi, B.A.; Prahara, A. Guava Fruit Detection and Classification Using Mask Region-Based Convolutional Neural Network. *Bul. Ilm. Sarj. Tek. Elektro* 2022, *4*, 186–193.

- 35. Xie, J.; Peng, J.; Wang, J.; Chen, B.; Jing, T.; Sun, D.; Gao, P.; Wang, W.; Lu, J.; Yetan, R.; et al. Litchi Detection in a Complex Natural Environment Using the YOLOv5-Litchi Model. *Agronomy* **2022**, *12*, 3054. [CrossRef]
- Yuan, T.; Lv, L.; Zhang, F.; Fu, J.; Gao, J.; Zhang, J.; Li, W.; Zhang, C. Robust Cherry Tomatoes Detection Algorithm in Greenhouse Scene Based on SSD. Agriculture 2020, 10, 160. [CrossRef]
- Coll-Ribes, G.; Torres-Rodríguez, I.J.; Grau, A.; Guerra, E.; Sanfeliu, A. Accurate detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and CNN methods. *Comput. Electron. Agric.* 2023, 215, 108362. [CrossRef]
- Qin, Z.; Xiaoliang, Y.; Bin, L.; Xianping, J.; Zheng, X.; Can, X.U. Motion planning of picking manipulator based CTB\_RRT\* algorithm. *Trans. Chin. Soc. Agric. Mach.* 2021, 52, 129–136.
- 39. Lin, G.; Zhu, L.; Li, J.; Zou, X.; Tang, Y. Collison-free path planning for guava-harvesting robot based on recurrent deep reinforcement learning. *Comput. Electron. Agric. Eng.* 2017, 33, 55–62.
- 40. Kothiyal, S. Perception Based UAV Path Planning for Fruit Harvesting; John Hopkins University: Baltimore, MD, USA, 2021.
- 41. Wecheng, W.; Gege, Z.; Xinlin, C.; Weixian, W. Research on path planning of orchard spraying robot based on improved RRT algorithm. In Proceedings of the 2nd International Conference on Big Data and Artificial Intelligence, Manchester, UK, 15–17 October 2020.
- 42. Kurtser, P.; Edan, Y. Planning the sequence of tasks for harvesting robots. Robot. Auton. Syst. 2020, 131, 103591. [CrossRef]
- 43. Liu, Y.; Qingyong, Z.; Yu, L. Picking robot path planning based on improved any colony algorithm. In Proceedings of the 34th Youth Acedemic Annual Conference of Chinese Association of Automation, Jinzhou, China, 6–8 June 2019.
- 44. Sarabu, H.; Ahlin, K.; Hu, A.-P. Graph-based cooperative robot path planning in agricultural environments. In Proceedings of the International Conference on Advanced Intelligent Mechatronics, Hong Kong, China, 8–12 July 2019.
- 45. Wang, L.; Wu, Q.; Lin, F.; Li, S.; Chen, D. A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robotic manipulators. *IEEE Access* 2019, *7*, 154332–154645.
- 46. Magalhaes, S.A.; Santos, F.N.D.; Martins, R.; Rocha, L.; Brito, J. Path planning algorithms benchmarking for grapevines pruning and monitoring. *Prog. Artif. Intell.* **2019**, *11805*, 295–306.
- 47. Zhang, K.; Lammers, K.; Chu, P.; Li, Z.; Lu, R. An automated apple harvesting robot—From system design to field evaluation. *J. Field Robot.* **2023**, *40*, 1–17. [CrossRef]
- 48. Hu, X.; Yu, H.; Lv, S.; Wu, J. Design and experiment of a new citrus harvesting robot. In Proceedings of the 2021 International Conference on Control Science and Electric Power Systems (CSEPS), Shanghai, China, 28–30 May 2021.

**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.